NAT'L INST. OF STAND & TECH R.I.C.
A11103 480615

NIST
PUBLICATIONS

NISTIR 4316

A Collection of Technical Studies Completed for the Computer-Aided Acquisition and Logistic Support (CALS) Program Fiscal Year 1988 Volume 2 of 3, Graphics, CGM MIL SPEC

Roy S. Morgan
Editor

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards
and Technology
National Computer Systems Laboratory
Gaithersburg, MD 20899

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
John W. Lyons, Director

QC
100
.U56
#4316
1990
C.2

NIST

NISTIR 4316

A Collection of Technical Studies Completed for the Computer-Aided Acquisition and Logistic Support (CALS) Program Fiscal Year 1988 Volume 2 of 3, Graphics, CGM MIL SPEC

Roy S. Morgan
Editor

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards
and Technology
National Computer Systems Laboratory
Gaithersburg, MD 20899

April 1990

Issued March 1991

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
John W. Lyons, Director

Executive Summary

The overall objective of the Department of Defense Computer-aided Acquisition and Logistic Support (CALS) Program is to integrate the design, manufacturing, and logistic functions through the efficient application of computer technology. CALS is a program to apply existing and emerging communications and computer-aided technologies in DoD and industry to:

- Integrate and improve design, manufacturing, and logistic functions; thereby bridging existing "islands of automation."
- Actively influence the design process to produce weapon systems that are more reliable and easier to support and maintain.
- Shift from current paper-intensive weapon support processes to a highly automated mode of operation, based on a unified DoD interface with industry for exchange of logistic technical information in digital form.

The CALS program was established by the Deputy Secretary of Defense in September 1985 to implement the recommendations of a Joint Industry/DoD Task Force. Management is provided by a DoD Steering Group, the OSD CALS Office, and a lead organization in each Military Department and the Defense Logistics Agency. The DoD CALS Office has obtained the support of the National Institute of Standards and Technology in the selection and implementation of CALS standards. An Industry Steering Group has also been established to focus the work of key industrial associations and the defense contractor community in CALS implementation.

The CALS strategy provides a plan for phased implementation of CALS. Phase I will apply current computer technology in existing/emerging DoD and industry systems for key logistic and design applications. Phase II will involve broad-based DoD and industry system redesign to implement advanced technology across a wider range of applications during the early 1990's. DoD is currently developing core Phase I requirements. Demonstrations and prototypes will support Phase I implementation, while advanced technology R&D continues for Phase II.

Implementation of the CALS program will result in:

- Design of more supportable weapon systems.
- Increased productivity and reduced cost of weapon system acquisition and logistic support.
- Improved timeliness and accuracy of logistic technical information.
- Enhanced operational readiness of military forces.

During FY86 NIST recommended standards to OSD which would be applicable to the DoD environment.[1] These recommendations included CALS use of standards in the areas of product definition, graphics, text, and data management.

CALS support work in FY87 focussed on the following activities: Developing a CALS framework, Development Plan and Core Requirements package; providing technical support for standards development and implementation; and conducting workshops and meetings to promote dialogue with the Services, the Defense Logistic Agency, and industry. A major thrust was the completion of the initial documentation of the high-priority standards required for CALS implementation.[2]

During FY88, a number of efforts advanced the development of technology and standards in support of CALS. These efforts were organized into the areas of Text, Graphics, and Product Data.

Text: Work on text and graphics standards in the CALS publishing environment included technology assessments, development of application guidance, conformance test plans and a draft FIPS for ODA/ODIF. Additionally, a technology assessment and proposed conformance testing strategy were developed for page description languages.

Graphics: The CALS efforts in CGM were continued and included work in the graphics standards committees and the expansion and updating of the CALS CGM Application Profile. The Application Profile was developed into a draft military specification. The draft was carried through the needed review and comment process and was published as MIL-D-28003 in December 1988. In addition, work on Extended CGM, or CGEM for CALS application was initiated. Work in

---

[1]Kemmerer, S., Editor, "Final NBS Report for CALS, FY86," U.S. Department of Commerce, National Bureau of Standards, NBSIR 87-3566, May 1987.

[2]Kemmerer, S. Editor, "A Collection of Technical Studies Completed for the Computer-aided Acquisition and Logistic Support (CALS) Program, Fiscal Year 1987," U.S. Department of Commerce, National Bureau of Standards, NBSIR 88-3726, NBSIR 88-3727, NBSIR 88-3728, and NBSIR 88-3729, March 1988.

the area of raster graphics continued. A draft MILSPEC for raster was developed which was later published as MIL-R-28002. The need for standards for the interchange of large format tiled raster documents was identified, and related technical papers were published separately as an NIST Internal Report.[3]

Product Data: The use of the Information Resource Dictionary System, (IRDS, ANSI Standard X3.138-1988) was proposed as an integration and configuration management mechanism for the Product Data Exchange Specification (PDES).

---

[3]Spielman, F., Editor, "Standards for the Interchange of Large Format Tiled Raster Documents," U.S. Department of Commerce, National Bureau of Standards, NBSIR 88-4017, December 1988.

These three volumes are a collection of the final reports presented to the DoD CALS Office.[4] The collection is divided as follows:

VOLUME 1.

Text, Security, and Data Management

Text and Graphics Standards in the CALS Publishing Environment

ODA/ODIF Application Guidance

Federal Information Processing Standards Publication (Draft) on Document Application Profile for the Office Document Architecture (ODA) and Interchange Format Standard

ODA/ODIF Conformance Test Plan

PDLs: A Technology Assessment

SPDL Conformance Strategy

Security

Risk Management Tools: A Guide to Selection and Use

Computer Security Issues in the Application of New and Emerging Information Technologies

Data Management

Information Resource Dictionary System: An Integration Mechanism for Product Data Exchange Specification

Using the Information Resource Dictionary System for PDES

VOLUME 2:

Graphics, CGM MIL-SPEC

CGM Conformance Testing

Final Phase I.1 CGM MILSPEC

Extended CGM MILSPEC Planning

---

[4]The publishing of this collection of reports does not imply that the CALS Office has endorsed the conclusions or recommendations presented.

VOLUME 3:

Graphics, CGM Registration

CGM Registration in Support of CALS

The following additional publications were completed by NIST during FY87 under separate cover. They are available through NTIS.

CALS Workshop Proceedings: CALS EXPO '88 'Quality and Productivity Through Integration" A DoD/Industry NIST Conference 4-6 October 1988

MIL-HDBK-59, Military Handbook: Department of Defense Computer-aided Acquisition and Logistic Support (CALS) Program Implementation Guide

MIL-STD-1840A, Automated Interchange of Technical Information

MIL-D-28000, Military Specification: Digital Representation for Communication of Product Data: IGES Application Subsets

MIL-M-28001, Military Specification: Markup Requirements and Generic Style Specification for Electronic Printed Output and Exchange of Text

MIL-R-28002, Military Specification: Raster Graphics Representation in Binary Format, Requirements for

MIL-D-28003, Military Specification: Digital Representation for Communication of Illustration Data: CGM Application Profile

CONTRIBUTIONS

NIST would like to acknowledge the major technical contributors to this volume. In alphabetical order they are:

Daniel Benigni

Peter Bono

George Carson

Lofton Henderson

GRAPHICS, CGM MIL-SPEC

CGM Conformance Testing

CALS SOW TASK 3.1.1

FINAL REPORT

CALS SOW TASK 3.1.1

CGM CONFORMANCE TESTING

TABLE OF CONTENTS

I. PURPOSE 1

II. BACKGROUND 1
1.0 Conformance Provisions of FIPS PUB 128, the CGM Standard 1
2.0 FY '87 CALS Accomplishments 1
3.0 Reasons for the CGM Application Profile 2
4.0 Need for Testing to FIPS PUB 128 and to MIL-D-28003 2
4.1 International Testing Efforts 3
4.2 FIPS Testing Efforts 3

III. DISCUSSION 4
1.0 Objectives 4
2.0 CALS Requirements 4
2.1 Review of CALS-Related Requirements for Standards 4
2.2 Role of the CGM in CALS 5
2.2.1 The Computer Graphics Metafile 5
2.2.2 The CALS Application Profile for CGM . 6
3.0 Commercial Implementations of the CGM 6
3.1 Overview 6
3.2 Product Analyses 7
3.2.1 Advanced Technology Center 7
3.2.2 AutoCAD via Zenographics Metafile Translator 8
3.2.3 Computer Associates International SuperImage 8
3.2.4 Computer Associates International (CAI) CGM Generator Library 8
3.2.5 Genographics 9
3.2.6 Grafpoint 9
3.2.7 Graphic Software System GSS*CGM . . . 10
3.2.8 Hewlett-Packard 11
3.2.9 IBM 12
3.2.10 Lotus Development Corporation 12
3.2.11 McDonnell Douglas CGM Toolkit 13
3.2.12 Motorola 14
3.2.13 Nova Graphics International 14
3.2.14 Pansophic StudioWorks version 3.00 . 15
3.2.15 Pansophic System, Inc. D-PICT/GKS . . 15
3.2.16 Precision Visuals (PVI) 16
3.2.17 Prime Computer 17
3.2.18 Software Publishing Corporation Harvard Graphics 17

TABLE OF CONTENTS (Continued)

3.2.19 Sun Microsystems 17
3.2.20 System One Software 18
3.2.21 Teknigraphics 20
3.2.22 Wasatch Computer Technology 20
3.2.23 Zenographics Mirage, Pixie, et al. . 21
4.0 Commercial Implementations .vs.Requirements of MIL-D-28003 21
4.1 Overview 21
4.2 Specific Requirements on Generators 22
4.2.1.1 Delimiter Elements 22
4.2.1.2 Metafile Descriptor Element Constraints 22
4.2.1.5 Graphical Primitives 24
4.2.1.6 Attribute Elements 24
4.2.1.7 Escape Elements 26
4.2.1.8 External Elements 26
4.2.2.1 Additional Linetypes 26
4.2.2.2 Additional Hatch Styles 26
4.3 Other Requirements 26
4.3.1 Encoding Format 26
4.3.2 Physical File Structure 27
4.3.3 Default Text Precision 27
4.3.4 Default Color Table 27
4.3.5 Fonts 27
4.3.6 Metafile Defaults Replacement 28
4.3.7 Handling of Out-of-range Parameter Values 28
4.3.8 Maximum List Sizes and Lengths 28
5.0 Conformance Testing 29
5.1 Levels of Testing 29
5.2 Overview of Required Tasks 30
5.2.1 Scope of the Testing Effort 31
5.2.2 Initial Tasks Associated With Certification Body Responsibilities . . 32
5.2.3 Initial Tasks Associated With Testing Laboratory Operations 35
5.2.4 Initial Tasks Associated With Client Responsibilities 37
5.3 Detailed Task Descriptions 38
5.3.1 Task Notation 38
5.3.2 Testing for Conforming CGM Generators 38
5.3.3 Testing for Conforming CGM Interpreters 47

TABLE OF CONTENTS (Continued)

IV. SUMMARY AND CONCLUSIONS 58
1.0 Summary of Commercial Implementations 58
2.0 Conformance Checklist Summary--Commercial .vs. MIL-D-28003 59
3.0 Summary of Conformance Testing Tasks 73
3.1 Development Phases 73
3.2 Summary Tables 75

V. ASSESSMENT OF IMPACT ON CGM TESTING 83
1.0 Impact on the CALS Program 83
2.0 Impact on the Commercial Marketplace 83
2.1 Constraints on Element Usage and on Options . 84
2.2 The Color Table and Fonts 84
2.3 Parameter Size Limits 85

2.4 Physical File Structure 85
3.0 Summary 86

VI. RECOMMENDATIONS 87
1.0 For CALS Certification of CGMs 87
2.0 Conforming Basic Generator 87
3.0 Conforming Basic Interpreter 88
4.0 Additional Remarks Regarding Certification 88

VII. REFERENCES 90

CGM CONFORMANCE TESTING

I. PURPOSE

Accelerate development of CGM (Computer Graphics Metafile) validation routines and ensure the input of CALS requirements in national and international standards processes. Prepare a plan and recommendation for certification of a testing laboratory (CALS SOW Task 3.1.1).

II. BACKGROUND

1.0 Conformance Provisions of FIPS PUB 128, the CGM Standard

The conformance statements in the CGM standard relate to the conformance of a metafile. They do not refer to the conformance of the generator or interpreter. There can be no expectation that a metafile sent to an unknown interpreter will be understood by that interpreter. Groups of users, such as CALS and MAP/TOP users, are concerned about this, and are trying to reduce the problem.

The CGM standard defines two levels of conformance: full conformance and functional conformance. Full conformance occurs when a metafile conforms to the abstract functional specification of Part 1 of the CGM standard, and also uses one of the three standard encodings. Functional conformance of a metafile occurs when a metafile conforms to the abstract functional specification, but a private encoding is used.

Thus, the standard is very limited in its conformance requirements. As cautioned in previous reports, and it still holds true--when CGM software is purchased, there can be no guarantee of minimum support by generator or interpreter software.

2.0 FY '87 CALS Accomplishments

NIST/NCSL (National Institute of Standards and Technology/National Computer Systems Laboratory--formerly NBS) was able to set direction for an architecture for the development of CGM conformance tests to coincide exactly with recommendations to CALS. First was that testing to the standard itself is not enough; the tests must include the CGM generators and interpreters. Part of the work last fiscal year was to develop a plan for the development of additional conformance tests needed to validate software that generates and reads

metafiles, in the form of a reference implementation for CGM. The approach taken in defining these tests has been to develop a plan for a reference implementation for metafile generators and interpreters, or a piece of software capable of generating any legal metafile and capable of interpreting any legal CGM, including testing for the CALS Application Profile (AP). In particular, one of last year's reports provides a functional specification and conceptual design for this reference implementation, as well as how it might be used as a basis for CGM testing tools or as a model for a CGM test service.

Second, the international test centres initially agreed to test to CGM Application Profiles, a concept that NIST/NCSL introduced at a key workshop in England. NIST/NCSL defined a CGM Application Profile for CALS, which became the basis for MIL-D-28003 (formerly MIL-D-CGM). These terms have been used interchangeably to mean the same thing; just remember that MIL-D-28003 is the most current designation.

3.0 Reasons for the CGM Application Profile

The CGM standard offers a useful method for the storage of graphical images. It defines a wide range of options which can be used by the generating software. These options include, for example, the precision of the data which is stored and the way that color is defined within the metafile. As stated above there is, however, no guarantee that the interpreting software will be able to make any sense of the metafile. The standard does not specify the behavior of generators and interpreters, and this makes it difficult for the purchaser of CGM software to guarantee that the software for generating and interpreting metafiles is what is required. Application profiles, such as the one designed for CALS, attempt to define the use of the standard. Those parties who use the CALS Application Profile should be able to predict the behavior of the generator and interpreter software. The metafiles written by one CALS application can thus be assured to be understood when transferred within the CALS environment.

4.0 Need for Testing to FIPS PUB 128 and to MIL-D-28003

It is important to ensure that the CALS Application Profile is adhered to by software purchased for the CALS effort. Therefore, it is necessary to offer some form of testing service for software to ensure that software does conform to the standard and to the CALS Application Profile.

Testing is important for ensuring that implementations do conform to the standards. Using existing testing methods it is impossible to guarantee that there are no errors in a product.

The testing strategy usually adopted attempts to show the presence of errors in the product. If a suitably large number of test cases is used, then confidence can be built in a product which handles these tests.

Existing validation suites adopt this philosophy and use a black box approach to testing; that is, the external specification or interface specifications of the product are examined and test cases generated, but no information is required about the internal workings of the implementation being tested.

Exhaustive testing is ideal but may be uneconomic to achieve. The best that can be achieved is to select a wide variety of test cases to exercise the implementation under test as fully as possible. It is important to ensure that the test cases generated have a high probability of detecting any errors in the implementation.

4.1 International Testing Efforts

NIST is also pursuing a standard at the international level for conformance testing of implementations of graphics standards (Reference 9). This standard is still in early draft stage, but it will specify a methodology for testing conformance to computer graphics standards of products which claim to implement these standards. This standard will directly address test requirements, test specifications, test suite, test procedures, and certification mechanisms.

4.2 FIPS Testing Efforts

In addition, NIST/NCSL has the goal of providing an initial structure and approach to uniform conformance testing programs Government-wide for appropriate FIPS Publications. The current initiative is the development and review of a draft FIPS (Reference 8) on conformance testing which will provide policy and guidelines for individual FIPS conformance testing program development.

The overall NCSL conformance program and an individual conformance testing approach for any given standard, will be continually developed as various testing procedures are prototyped; test cases are gathered, studied, and evaluated; industry and other national European programs are studied; and experience is gained. This evolution will also apply to the various conformance testing programs for graphics. Any conformance testing program that NIST/NCSL advocates for use by DOD CALS will be based on this approach to testing being developed. In the case of the CGM standard, this will also mean adding conformance testing to test conformance to MIL-D-28003.

III. DISCUSSION

1.0 Objectives

The objectives of this task were to:

1. Perform a study of the variability of commercial CGM generator implementations, including the degree of conformance to the CALS Application Profile for CGM. Note that this comparison is made with the April 1988 version of this specification. These commercial implementations are analyzed in section 3 below, compared to MIL-D-28003 in section 4 below, and both aspects summarized in sections IV, subheadings 1 and 2.

2. Assess the impact of CGM test development strategy with respect to the CALS environment and the marketplace. This is accomplished in section V.

3. Provide input and recommendations for a plan for CALS on how certification of CGMs to the Application Profile could be handled. This involves preparing a list of tasks that must be accomplished in order to put in place a testing service (both for FIPS PUB 128 and MIL-D-28003) that is consistent with:

   a. The draft proposed FIPS on Conformance Testing Policy and Procedures (Reference 8), and

   b. The working Draft International Standard (DIS), "Conformance Testing of Implementations of Graphics Standards (Reference 9).

   These testing tasks are developed in section 5 below, and summarized in section V, subheading section 3.

2.0 CALS Requirements

2.1 Review of CALS-Related Requirements for Standards

References 1 and 2 contain analyses of CALS requirements for graphics-related standards in the areas of engineering design, technical publishing, procurement support, and interactive delivery systems.

This report focusses on the picture interchange requirements of CALS when applied to the task of technical manual publishing and illustration.

2.2 Role of the CGM in CALS

2.2.1 The Computer Graphics Metafile

The CGM provides a file format suitable for the storage and retrieval of picture description information. The file format consists of an ordered set of elements that can be used to describe pictures in a completely device-independent way. One or more pictures can be stored in a single metafile, and the metafile is defined in such a way that, in addition to sequential access to the whole metafile, random access to individual pictures is well defined. That is, the pictures are completely independent, one from another: their appearance does not depend upon the order in which they are accessed or displayed.

In addition to a functional specification, the CGM standard documents three standard encodings of the metafile semantics. The Character encoding requires minimum metafile size and is suitable for transmission across networks of heterogenous systems but is expensive to encode and decode. The Binary encoding requires minimum effort to generate and interpret but is not well-suited for exchange between computers of different arithmetic data types. It is nearly as efficiently coded as the Character encoding. The Clear-text encoding provides maximum readability and editability for ease of use by humans (e.g., for debugging purposes) but, generally, pays a heavy penalty in size and performance. The size is much larger because English and other natural languages contain a lot of redundancy. The performance is worse because parsing and recognizing text strings and converting text strings to internal numbers for use by a graphics subsystem is expensive in its use of CPU cycles.

In reference 1, the standardized CGM elements are listed by type. The ESCAPE and APPLICATION DATA elements have been provided to support uses of the CGM in ways that go beyond the exchange of pictures. Nongraphical data and graphical elements not yet standardized can be incorporated into metafiles in a regular way. When these extended metafiles are exchanged by cooperating processes, standard commercial products can be used to handle the standard metafile elements, and new code need be written only for the special, non-standardized elements. Large groups of users of extended metafiles can get together and agree upon a set of extensions--just like MAP and TOP users have agreed upon guidelines to the implementation of the OSI standards. For example, the elements of a business chart--like legend entries, tick marks, and axis labels--or the elements of a project schedule--like PERT chart symbols, milestone markers, or title-could be marked in the metafile. An editing program could be written to read such metafiles and allow modifications to them before rendering the chart on a hardcopy device or including it in a report or manual.

In the absence of any facsimile standard capable of handling multicolor images (i.e., those with more than one bit per pixel), a CGM employing only the CELL ARRAY primitive could be used. Images expressed with either indexed and direct color specifications can be represented. In the Character-Coded and Binary encodings, run-length encoding may be used to reduce the size of the resulting CGM files.

2.2.2 The CALS Application Profile for CGM

Reference 5, MIL-D-28003, specifies a set of conditions to be met by CGM generators and interpreters if they are to be said to be "conforming" to the CALS Application Profile for CGM. This report is particularly interested in those requirements that must be met for an implementation to be certified as a CALS "conforming basic generator" (see 3.1.1, Reference 5).

The specific requirements that must be met are elaborated in detail in section 5.3.2 below.

3.0 Commercial Implementations of the CGM

3.1 Overview

The objective was to determine to what degree current commercial implementations of CGM generators adhere to the requirements of MIL-D-28003 for conforming basic generators.

[NOTE: This report is intended to be informative in terms of this comparison, and not a critical evaluation of any commercial software system. Inclusion of any software system in the following sections in no way implies a recommendation or endorsement by NIST or DoD, and the presentation should not be construed as a certification that any system does or does not provide the indicated capabilities. Further, if any software system has been omitted from this study, that does not imply that its capabilities are more or less than those of systems included herein.]

Most implementations of generators were developed with little or no knowledge of the CALS requirements for CGMs. Consequently, these implementations reflect the company's judgments as what features are needed in their commercial marketplace without regard for special requirements from DoD.

Over 100 metafiles from 24 products offered by 20 companies and organizations were examined. Many of the products share underlying CGM generation technology. Five of the products use some release of Graphics Software Systems' GSS*CGM; four are

based on the generator libraries licensed from Henderson Software; three on the GKS Metafile Output workstation of Advanced Technology Center's (ATC's) GRAFPAK-GKS; and two on Nova Graphic International's GKS CGM generation capability.

All CGMs analyzed followed the Binary Encoding.

Those CGMs based on technology from Henderson Software, ATC, and McDonnell Douglas are more likely to pay some attention to the specifications in the TOP Application Profile for CGM, precursor and model for the CALS Application Profile.

All of the metafiles were generated on either PCs or workstations. None of them were generated on Apple Macintoshes.

In section 3.2 below, an overview of the CGMs written by each of the various products is provided without explicit reference to the requirements of MIL-D-28003. Then, in section 4 below, to what degree the CGMs produced by each of these products conforms to the requirements of MIL-D-28003 is specifically evaluated.

3.2 Product Analyses

3.2.1 Advanced Technology Center

ATC has developed its own base technology as a GKS metafile output workstation driver, accessed through its GRAFPAK-GKS. The driver can generate multiple pictures per metafile.

The Metafile Descriptor sets 16-bit integer VDCs; 16-bit integer, index precision; and (16,16) fixed-point real precision. Colour and colour index precision is always 8-bits, and colour value extent is always 255.

There is no use of Metafile Defaults Replacement, Font List, Character Set List, and Character Coding Announcer.

There is no use of Scaling Mode, Colour Selection Mode, or any of the Specification Modes. Background Colour is used.

The generator does not use Auxiliary Colour, Transparency, and Clip Indicator. It does use Clip Rectangle.

The generator does not use Disjoint Polyline, Restricted and Append Text, Polygon Set, Circular Arc 3 Point and 3 Point Close. Polygons are limited to 500 vertices; Text is limited to 256 characters; only the "pie closure" forms of Circular Arc Center and Elliptical Arc Close are used.

Eleven GDPs with negative ids can be generated.

The generator does not use Character Set Index and Alternate Character Set Index nor the Edge attributes. It uses all other attribute elements.

Up to 10 line, marker, and text bundles may be referenced and up to 10 pattern table entries may be specified and referenced. Implementation dependent (negative valued) line, marker, and hatches may be generated. Implementation dependent fonts are specified with indices 11 through 18. Empty interior style is never generated. Standard hatch indices 5 and 6 cannot be generated.

Seventy ESCAPEs with negative ids can be generated.

MESSAGE and APPLICATION DATA elements can be generated.

3.2.2 AutoCAD via Zenographics Metafile Translator

AutoCAD DXF files can be translated to CGMs via a Zenographics product, called Metafile. See the Zenographics description (section 3.2.23).

The Metafile Elements List contains an enumerated set of 51 elements, which is a superset of the file contents.

The file defines color indexes 0-7 in a single COLOR TABLE element 8 RGB triples long: Black, Red, Yellow, Green, Cyan, Blue, Magenta, White.

3.2.3 Computer Associates International SuperImage

SuperImage uses the GSS*CGM metafile driver. See the Graphic Software Systems description (section 3.2.7) for the details.

Absolute line width is initially set to 1, but it can be changed later in the CGM.

3.2.4 Computer Associates International (CAI) CGM Generator Library

CAI uses base technology developed with the assistance of Henderson Software. This CGM generator was developed for use with ISSCO mainframe and minicomputer products like DISSPLA and TELL-A-PLAN.

The Metafile Descriptor is minimal. The METAFILE ELEMENT LIST contains DRAWING plus some individual elements beyond the DRAWING set. There is a METAFILE DEFAULTS REPLACEMENT element that sets VDC EXTENT to (0,0), (4095,4095).

The Picture Descriptor contains only SCALING MODE and VDC EXTENT. The VDC extent is set explicitly to roughly the aspect ratio of an 8.5 x 11 sheet (1144 by 884). The SCALING MODE of the original files ware METRIC, but the scale factor was not chosen for a reasonable size drawing. This element is scheduled for repair by CAI in the next product release.

3.2.5 Genographics

Genigraphics developed its own base technology.

The Metafile Descriptor includes all elements except METAFILE DEFAULTS REPLACEMENT, CHARACTER SET LIST, and CHARACTER CODING ANNOUNCER. The METAFILE ELEMENTS LIST enumerates a superset of the elements in the metafile. The FONT LIST contains one entry: "ISO646." All other descriptor elements have values corresponding to CGM defaults except:

COLOR PRECISION: 16

COLOR INDEX PRECISION: 16

MAXIMUM COLOR INDEX: 255.

The Picture Descriptor contains SCALING MODE and COLOR SELECTION MODE, setting CGM default values; LINE WIDTH SPECIFICATION MODE setting ABSOLUTE; and VDC EXTENT setting (332,0), (32435,24076).

Control elements present are VDC INTEGER PRECISION and TRANSPARENCY, setting CGM defaults, and CLIPPING INDICATOR setting ON.

Genigraphics uses a basic Primitive set of POLYLINE, TEXT, POLYGON, and RECTANGLE in all pictures of this set.

3.2.6 Grafpoint

Grafpoint developed its own base technology.

In the Metafile Descriptor VDC TYPE, INTEGER PRECISION, and INDEX PRECISION set values that correspond to the CGM defaults. COLOR PRECISION and COLOR INDEX PRECISION are set to 16 bits. REAL PRECISION is set to floating point; the files do not contain any real operands, so floating point is not required. COLOR VALUE EXTENT is set to 0-1000 in each of the R, G, and B components. The Metafile Elements List contains an enumerated set of 51 elements, which is a superset of file contents.

In the Picture Descriptor, SCALING MODE and COLOR SELECTION MODE elements set values corresponding to the CGM defaults. LINE WIDTH SPECIFICATION MODE and MARKER SIZE SPECIFICATION MODE are set to ABSOLUTE. VDC EXTENT is set to (0,0), (16383,16383).

The picture contains no control elements other than VDC INTEGER PRECISION, which contains the CGM default value.

The picture starts by defining color indexes 0-15 with a single COLOR TABLE element, then many of the entries are immediately redefined one at a time.

3.2.7 Graphic Software System GSS*CGM

GSS developed its own base technology as a CGI driver, accessed through its GSS*CGI.

GSS*CGM can generate multiple pictures per metafile.

VDCs are always 16-bit integers; Integer, Index, Colour and Colour Index Precision is always 16-bits; Real Precision is always fixed (16,16). Maximum Colour Index is always 255 and colour extents always range from 0 to 1000.

GSS*CGM does not use Metafile Defaults Replacement, Font List, Character Set List, and Character Coding Announcer.

Scaling Mode is "abstract;" Colour Selection Mode is "indexed;" and Line Width and Marker Size Specification Mode is "absolute." Edge Width Specification Mode is not used. Background Colour is initially set to (0,0,0) but can be changed by the generating program.

VDC EXTENT is always (-32768,-32768,32767,32767) for GSS*CGM versions 2.12 or earlier; VDC EXTENT is always (0,0,32767,32767) for version 2.13 or later. In both cases, actual VDCs contained in other metafile elements always lie in the positive (first) quadrant. This anomaly in early GSS*CGMs is known as the "1/4-frame problem." While GSS metafiles are not syntactically illegal, they do violate the spirit of the CGM standard in that the VDC EXTENT is supposed to represent the "region of interest" of the picture.

All the Control Elements except VDC Real Precision can be generated. VDC Integer Precision is always 16 bits.

Of the Graphical Primitives, only Disjoint Polyline, Restricted and Append Text, Polygon Set, GDP, and Circular Arc 3 Point and 3 Point Close cannot be generated.

Bundles and ASFs are not used nor are the Edge attributes. Of the text attributes, Character Spacing, Character Expansion Factor, Text Precision, and Text Path cannot be generated. Of the remaining attributes, only Interior Style, Fill Colour, Hatch and Pattern Index, and Color Table are used.

Font indices 1 through 100 are meant to refer to hardware fonts; font indices 101 through 106 select six Hershey software fonts as follows: Simplex, Complex, Complex Italic, Duplex, Triplex, and Triplex Italic.

Private attributes include Line Type -1 for a "medium dashed" line; Marker Type -1 for a "diamond"; and Hatch Indices -1 (medium-spaced +45 degree lines), -2 (widely-spaced +45 degree lines), -3 (medium-spaced +45 and -45 degree lines), and -4 (widely-spaced +45 and -45 degree lines).

3.2.8 Hewlett-Packard

Hewlett-Packard uses base technology developed with the assistance of Henderson Software and accessed through HP's STARBASE software.

HP STARBASE can generate multiple pictures per metafile. HP also claims to be able to generate metafiles that conform to the TOP/BASIC application profile.

The Metafile Descriptor contains most of the CGM descriptor elements, lacking only FONT LIST and CHARACTER SET LIST. However, most of these appear with values corresponding to CGM defaults. VDCs are always integer, but a variety of precisions can be generated. Likewise, the other Class 1 elements can be generated with a variety of settings.

Most files contain a METAFILE DEFAULTS REPLACEMENT element, which sets the VDC INTEGER PRECISION to 16 bits and the VDC EXTENT to (-32768,-32768), (32767,32767).

There is no use of the Edge Width Specification Mode. Background Colour is used.

The generator does not use Auxiliary Colour and Transparency. It does use Clip Rectangle and Clip Indicator.

The generator does not use Disjoint Polyline, Restricted and Append Text, Cell Array, Rectangle, Circle, Circular Arc 3 Point and 3 Point Close, Circular Arc Centre and Centre Close, and the elliptical elements.

Polylines and Polymarkers are broken into groups of no more than

1024 vertices; Polygons and Polygon Sets are potentially unlimited in size; Text is limited to 255 characters.

No GDPs can be generated.

The generator does not use Line, Marker, Text, Fill, and Edge Bundles, nor ASFs; Line Width; Character Set Index and Alternate Character Set Index; interior style "hatch" or "pattern"; any of the Edge attributes; Fill Reference Point; Pattern Table and Size. It uses all other attribute elements.

Implementation dependent (negative valued) line types and markers may be generated. Implementation dependent fonts are specified with positive-valued indices. Only "hollow" and "solid" interior style can be generated.

ESCAPE elements with negative ids be generated.

MESSAGE and APPLICATION DATA elements can be generated.

3.2.9 IBM

The IBM Graphics Development Toolkit can use GSS*CGM metafile driver through the IBM Virtual Device Interface (VDI). See the GSS*CGM summary (section 3.2.7) for a complete discussion, including description of the "1/4 frame problem."

Another syntax error is present in some of the files: the data associated with ESCAPE elements is incorrectly coded. The data are supposed to be coded as string data type, but the string length field is missing from the data stored after the ESCAPE identifier field.

3.2.10 Lotus Development Corporation

Lotus developed its own base technology. There is only one picture per metafile; it is always called "PICTURE 1".

The Metafile Descriptor contains nine of the CGM Metafile Descriptor elements. The Metafile Descriptor specifies 16-bit integer VDCs; 16-bit integer, index precision; (16,16) fixed-point real precision; colour precision is not used; colour index precision is always 16-bits; maximum colour index is always 16 and colour value extent is not specified. Only COLOR INDEX PRECISION (16) and MAXIMUM COLOR INDEX (16) have non-default values. There is no use of Metafile Defaults Replacement, Font List, Character Set List, and Character Coding Announcer. The METAFILE ELEMENT LIST contains an enumerated superset of the elements in the file.

The Picture Descriptor sets VDC EXTENT to a maximal horizontal rectangle inscribed in the 0-32K square. Scaling Mode is always "abstract," Colour Selection Mode always "indexed," and Line Width and Marker Size Specification Mode always "absolute." Edge Width Specification Mode and Background Colour are not used.

The generator does not use Auxiliary Colour or Clip Rectangle. Transparency is always "on" and Clip Indicator is always "off."

The generator does not use Disjoint Polyline, Restricted and Append Text, Polygon Set, Cell Array, Circular Arc 3 Point and 3 Point Close.

The generator does not use Line, Marker, Text, Fill, and Edge Bundles, nor ASFs; Text Precision and Path; Character Expansion Factor and Spacing; Character Set Index and Alternate Character Set Index; interior style "hatch" or "pattern"; any of the Edge attributes; Fill Reference Point; Pattern Table and Size. It uses all other attribute elements.

The Colour Table element is not used, but up to 12 colour indices are used (with a built-in predefined colour table assumed).

Implementation dependent (negative valued) line, marker, and hatches may be generated. Line widths take on only five values: 1, 20, 80, 175, and 275. Implementation dependent fonts are specified with indices 1 through 8. Empty interior style is never generated. One private line type (-1=medium dash) can be generated. Four illegal marker types (6=dash; 7=filled circle; 8=filled square; 9=filled rectangle) and one private marker value (-1=diamond) can be generated. Four private hatch values (-1 through -4) and 6 illegal hatch values (10-15 for gray scale fills) can be generated.

No GDPs are generated and one ESCAPE with an id of -201 (Set Writing Mode) can be generated. No MESSAGE elements but two APPLICATION DATA elements, 16975 and 17743, are used to flag the beginning and end of closed objects defined by filled objects and their outlines.

3.2.11 McDonnell Douglas CGM Toolkit

McDonnell Douglas developed its own base technology.

The Metafile Descriptor is close to that specified for a TOP metafile. REAL PRECISION is floating point (9,23). The METAFILE DEFAULTS REPLACEMENT sets: VDC REAL PRECISION to floating point (9,23); TEXT PRECISION to STROKE; and color table entries 2-9 as per the TOP specification (Red, Green, Blue, Yellow, Magenta, Cyan, Black, White). There is a FONT LIST with 4 Hershey fonts as per the TOP application profile. The VDC TYPE is set to REAL

by the last element in the metafile descriptor. While this is legal, it is unusual to have it follow the VDC REAL PRECISION setting in the Metafile Default Replacements element.

The file uses DIRECT COLOR SELECTION MODE (but there is a COLOR TABLE in the picture anyway). The SCALING MODE is METRIC, with a scale factor of 25.4 (converting mm to inches). Together with VDC EXTENT of (0,0) to (11.5,8.5), this means a precisely scaled picture at normal (US) paper size. Immediately within the first picture body, VDC REAL PRECISION is set to fixed point (16,16). The coordinates in the picture body are fixed point, while those in the picture descriptor are floating point.

Generally speaking, McDonnell Douglas metafiles contain partitioned elements. Note: some files from this generator have contained zero-length partitions, which are legal but which some interpreters may not expect.

3.2.12 Motorola

Motorola uses base technology developed by Nova Graphics International. See the Nova Graphics summary (section 3.2.13) for more detail.

3.2.13 Nova Graphics International

Nova Graphics developed its own base technology as a GKS metafile output workstation driver, accessed through its NOVA*GKS.

The driver can generate multiple pictures per metafile.

It can produce both 16-bit and 32-bit integer and both (16,16) fixed point and (9,23) floating point real VDCs; 8, 16, and 32-bit integer, index precision; (16,16) fixed-point and (9,23) floating point real precision; colour and colour index precision can be 8, 16, or 32 bits; maximum colour index is always 255 and colour value extent is 0-255.

There is no use of the Character Set List and Character Coding Announcer elements.

From NOVA*GKS, Scaling Mode is always "abstract;" Colour Selection Mode is "indexed;" and the Specification Modes are always "scaled."

All CGM Control Elements can be generated.

All CGM Graphical Primitive Elements can be generated, although at present NOVA*GKS does not currently generate any GDP's.

All CGM Attribute Elements can be generated, although at present NOVA*GKS does not use Character Set Index and Alternate Character Set Index and does not generate "continuous alignment" values of the TEXT ALIGNMENT element.

Up to 20 line, marker, and text bundles may be referenced. Implementation dependent (negative valued) line, marker, and hatches are not generated. Implementation dependent fonts are specified with indices 1 through 15. Empty interior style is never generated. These standard limits are changeable by the NOVA*GKS system installation manager.

NOVA*GKS does not currently generate any CGM ESCAPE elements.

MESSAGE and APPLICATION DATA elements can be generated.

3.2.14 Pansophic StudioWorks version 3.00

Pansophic developed its own base technology for its artist workstation product line.

The Metafile Descriptor sets 16-bit integer VDCs; 16-bit integer, index precision; and (16,16) fixed-point real precision. Only direct colour of 8-bits precision is used; colour value extent is 0 to 255.

The generator makes no use of Metafile Defaults Replacement, Font List, Character Set List, and Character Coding Announcer.

The generator makes no use of Scaling Mode or Marker Size Specification Mode. Colour Selection Mode is "direct;" Line Width and Edge Width Specification Modes are always "absolute." Background Colour is used.

None of the Control elements are generated.

Only three Graphical Primitive elements are generated: Polyline, Text, and Polygon.

Only 10 of the 35 Attribute elements can be generated (with restrictions as described in the following): Line Type (always 1="solid"); Line Width; Line Colour; Text Colour; Character Height; Interior Style ("hollow" or "solid" only); Fill Colour; Edge Type (always 1="solid"); Edge Colour; and Edge Visibility (always "on").

ESCAPE, MESSAGE and APPLICATION DATA elements cannot be generated.

3.2.15 Pansophic System, Inc. D-PICT/GKS

Pansophic distributes ATC GRAFPAX*GKS and has used it to build certain layered application products in its D-PICT product line, such as D-PICT/GKS and D-PICT/INTELLICHART, a stand-alone presentation graphics program. It uses the base technology developed by ATC. See the ATC GRAFPAK*GKS description (section 3.2.1) for details.

3.2.16 Precision Visuals (PVI)

PVI uses base technology developed by Henderson Software. The behavior of the generator is controlled by a "software switch" which can take on one of three values:

PVI Mode
: PVI escape functions, special opcodes, and GDPs are converted to CGM escape and GDP elements. Private values may occur in the CGMs so generated.

MAP/TOP Conforming Mode
: PVI escape functions and GDPs are NOT mapped to CGM elements. However, other private values are mapped as in PVI mode.

MAP/TOP Basic Mode
: Follows the rules for Conforming Mode and, in addition, no private values (e.g., proprietary line types and hatch styles) are placed in the metafile.

The Metafile Descriptor is fairly minimal. It does contain a METAFILE DEFAULTS REPLACEMENT that sets color indexes 2-9. The METAFILE ELEMENT LIST consists of DRAWING and METAFILE DEFAULTS REPLACEMENT.

All default precisions and default colour value extent are assumed. There is no use of Font List, Character Set List, and Character Coding Announcer.

Of the Picture Descriptor and Control elements, only Colour Selection Mode, VDC Extent, and Background Colour can be generated.

The following Graphical Primitive Elements can be generated: Polyline, Polymarker, Polygon, and Text.

The following Attribute Elements can be generated: Line, Marker, Text, Fill, and Edge Colour; Line Type and Line Width; Marker Type; Text Font Index, Character Expansion Factor, Character Height, Character Spacing, Text Alignment, Text Precision; Interior Style and Hatch Index; Edge Visibility; and Colour Table.

Depending on Mode, ESCAPE, GDP, and APPLICATION DATA elements may be generated. MESSAGE cannot be generated.

3.2.17 Prime Computer

Prime uses base technology developed by ATC and can generate multiple pictures per metafile.

The Metafile Descriptor sets 16-bit integer VDCs; 16-bit integer, index precision; and (16,16) fixed-point real precision. Colour and colour index precision are always 8-bits; maximum colour index is not generated; the colour value extent is 0 to 255.

There is no use of Metafile Defaults Replacement and Font List. Character Set List and Character Coding Announcer are set up to select ISO 646 in 8-bit format.

Scaling Mode is always "abstract," Colour Selection Mode is "indexed," and Line Width and Marker Size Specification Modes are "scaled." Edge Width Specification Mode is not used. Background Colour is used.

The generator does not use Auxiliary Colour and Transparency but does use Clip Rectangle and Clip Indicator.

The generator does not use Disjoint Polyline, Restricted and Append Text, Polygon Set, Circular Arc 3 Point and 3 Point Close. Only the "pie closure" forms of Circular Arc Center and Elliptical Arc Close are used.

The generator uses all attribute elements, although the Character Set Index and Alternate Character Set Index are always set to 1 (referring to ISO 646).

GDP and ESCAPE elements with negative ids can be generated.

MESSAGE and APPLICATION DATA elements can be generated.

3.2.18 Software Publishing Corporation Harvard Graphics

Harvard Graphics is built on top of GSS*CGI. Consequently, it uses the GSS*CGM metafile driver. See the GSS description (section 3.2.7) for details.

Polylines and Polygons contain a maximum of 200 vertices. Text strings contain a maximum of 60 characters.

3.2.19 Sun Microsystems

Sun uses base technology developed with the assistance of Henderson Software and accessed through Sun Microsystem's SunGKS software.

SunGKS can generate multiple pictures per metafile.

The generator uses the CGM default precisions. Maximum Colour Index is 255. VDC type is integer at either 16-bit or 32-bit precision. Metafile Defaults Replacement establishes the VDC INTEGER PRECISION, Text Precision as "string," and loads the default TOP colour table of 254 elements, for indices 2 through 255.

Font List and Character Set List elements are not generated, but the Character Coding Announcer is set to "BASIC7BIT."

All the picture descriptor elements are generated. The VDC EXTENT is taken from the GKS Workstation Window.

The generator does not use VDC Real Precision, Auxiliary Colour, and Transparency. It does use Clip Rectangle and Clip Indicator.

Because it is positioned as a GKS metafile output workstation, the generator uses only five of the nineteen available graphical primitives: Polyline, Polymarker, Text, Polygon, and Cell Array.

Polyline, Polymarker, and Polygon are limited to 1024 vertices; Text is limited to 255 characters in TOP mode; 32,767 characters otherwise.

No GDPs can be generated.

The generator does not use Character Set Index and Alternate Character Set Index; interior style "empty"; any of the Edge attributes; and ASFs. It uses all other attribute elements.

Implementation dependent (negative valued) line types and markers may be generated. Implementation dependent fonts are specified with indices 1 through 16. A maximum of eight 32x32 colored patterns can be specified.

No ESCAPE elements can be generated.

MESSAGE but not APPLICATION DATA elements can be generated.

3.2.20 System One Software

System One Software developed its own base technology. The generator technology appears in Presentation Technologies

ImageMate and Kinetic Graphics Systems business graphics products.

Only one picture per metafile can be generated.

The Metafile Descriptor contains most of the CGM Metafile Descriptor elements.

The Metafile Descriptor sets 16-bit integer VDCs; 16-bit integer, index precision; and (16,16) fixed-point real precision. Colour precision is 8-bits; colour index precision is always 16-bits; maximum colour index is always 255; and colour value extents are 0 to 255 in R, G, and B. Most of these set values corresponding to the defaults of CGM. The exceptions are COLOR INDEX PRECISION (16) and MAXIMUM COLOR INDEX (255).

There is no use of Metafile Defaults Replacement.

There is a FONT LIST element with 2 entries, taken from the NCGA Integrate 88 demonstration:

NCGA GRAFNET:SERIF BOLD

NCGA GRAFNET:SERIF BOLD-ITALIC

There is also a CHARACTER CODING ANNOUNCER that sets EXTENDED 7-BIT. However, the coding technique is not relied upon in the metafile.

The Picture Descriptor contains all of the CGM Picture Descriptor elements. Scaling Mode is always "abstract," Colour Selection Mode always "indexed," and Line and Edge Width and Marker Size Specification Mode always "absolute."

The only Control element used is Transparency.

The generator does not use Cell Array. Polyline, Polymarker, and Polygon are limited to 370 vertices; Text is limited to 256 characters.

No GDPs are generated.

The generator does not use Character Set Index, Alternate Character Set Index, Pattern Table and Pattern Size. It uses all other attribute elements.

Implementation dependent fonts are specified with indices 1 and 2. The Font List element indicates which fonts are associated with the two indices.

ESCAPE elements with negative ids can be generated.

No MESSAGE elements but some APPLICATION DATA elements can be generated.

3.2.21 Teknigraphics

Teknigraphics metafiles are created by GRAPH-TEX, an application that captures a Tektronix TCS data stream and converts it to a CGM file, using a relatively early version of the GSS*CGM driver.

Its Metafile Descriptor and Picture Descriptor follow the GSS profile presented in the description for GSS*CGM (section 3.2.7).

3.2.22 Wasatch Computer Technology

Wasatch developed its own base technology.

The Metafile Descriptor contains:

METAFILE VERSION
METAFILE DESCRIPTION
VDC TYPE
INTEGER PRECISION
INDEX PRECISION
COLOR INDEX PRECISION
MAXIMUM COLOR INDEX
METAFILE ELEMENT LIST.

All of these set values corresponding to CGM defaults (where there are defaults specified in the standard), except that MAXIMUM COLOR INDEX is specified as 255 and the METAFILE ELEMENT LIST is specified as DRAWING-PLUS-CONTROL.

The Picture Descriptor contains:

COLOR SELECTION MODE
LINE WIDTH SPECIFICATION MODE
MARKER SIZE SPECIFICATION MODE
VDC EXTENT.

These also just set the values to the CGM defaults, except for VDC EXTENT.

There is a COLOR TABLE that sets 255 entries starting at index 2. This requires a color table to be 257 entries long, if the default (interpreter-dependent) values of indexes 0 and 1 are to be preserved. All these color tables have been patched so that only 254 entries are loaded (indices 2 through 255), consistent with the specification of MAXIMUM COLOR INDEX as 255.

3.2.23 Zenographics Mirage, Pixie, et al.

All Zenographics products use technology developed by Zenographics.

Only one picture per metafile can be generated. The Metafile Description does not identify the product name, but rather specifies the file name of the CGM.

Metafile descriptor elements select 16-bit integer VDCs; 16-bit integer, index precision; and (16,16) fixed-point real precision. Colour precision and colour index precision are always 16-bits; maximum colour index is always 255 and colour value extents are 0 to 1000.

The generator makes no use of Metafile Defaults Replacement, Font List, Character Set List, and Character Coding Announcer.

Scaling Mode is always "abstract," Colour Selection Mode always "indexed," and Line Width always "absolute." Marker Size Specification Mode and Edge Width Specification Mode are not used.

None of the Control elements are used.

The generator does not use Restricted and Append Text, Polygon Set, Cell Array, Circular Arc 3 Point and 3 Point Close, Ellipse and Elliptical Arc. Polyline, Polymarker, and Polygon are limited to 480 vertices.

No GDPs are generated.

The generator does not use Line, Marker, Text, Fill, and Edge Bundles, nor ASFs; the Marker attributes; Text Precision and Path; Character Set Index and Alternate Character Set Index; interior style "hatch," "pattern", and "empty;" Fill Reference Point; Pattern Table and Size. It uses all other attribute elements.

No ESCAPE, MESSAGE, or APPLICATION DATA elements can be generated.

4.0 Commercial Implementations .vs.Requirements of MIL-D-28003

4.1 Overview

MIL-D-28003 (Reference 5) establishes the requirements to be met when two-dimensional picture description or illustration data is delivered in the digital format of the Computer Graphics Metafile (CGM) as specified in FIPS PUB 128.

MIL-D-28003 provides an Application Profile for metafiles, generators, and interpreters. Only one level of conformance is specified for metafiles and generators; two levels (draft and publication) are specified for interpreters.

This study is interested in the requirements that must be met by CGM generators in order to be deemed "conforming basic generators." The next section abstracts from MIL-D-28003 these requirements and discusses which CGM implementations fail to meet these requirements. All of the specific requirements are organized by their MIL-D-28003 paragraph numbers, except that specific requirements in this document are in sections numbered 4.2.x.y, while those in MIL-D-28003 will be in the section 3.2.x.y. Each requirement is stated, then an analysis of the current state-of-the-art is presented. (Note that, as a consequence of the paragraph numbering, not all sub-sub-paragraph numbers will necessarily be represented within sub-paragraph 4.2.)

Other requirements, drawn from other paragraphs of MIL-D-28003 are presented and discussed in section 4.3.

4.2 Specific Requirements on Generators

4.2.1.1 Delimiter Elements

No-ops are limited in size to no more than 32767 octets.

Analysis: No CGM observed violates this limit.

4.2.1.2 Metafile Descriptor Element Constraints

a1. Metafile Description shall include a substring briefly identifying company or product.

Analysis: Most CGMs meet this requirement; however, those of ATC, Pansophic (only its D-PICT/GKS), and Zenographics are currently deficient in this respect. This requirement is easy to meet.

a2. Metafile Description shall contain the substring "MIL-D-28003/BASIC-1."

Analysis: None of the implementations have had time to meet this requirement; however, it is extremely simple to meet. Furthermore, ATC, Genigraphics, Hewlett-Packard, McDonnell Douglas, Precision Visuals, and Sun all have a mode modelled after "TOP/BASIC" and contain such a string within their Metafile Description. This shows

these companies are at least aware of the concepts that gave rise to the CALS application profile.

b. Integer Precision shall be 16.

Analysis: No implementation fails to meet this requirement, although Nova does permit CGMs to be generated with 32-bit precision.

c. Real Precision shall be either (1,16,16) or (0,9,23).

Analysis: No implementation fails to meet this requirement.

d. Index Precision shall be 16.

Analysis: No implementation fails to meet this requirement, although Nova does permit CGMs to be generated with 32-bit precision.

e. Color Precision shall be either 8 or 16.

Analysis: No implementation fails to meet this requirement, although Nova does permit CGMs to be generated with 32-bit precision.

f. Color Index Precision shall be either 8 or 16.

Analysis: No implementation fails to meet this requirement, although Nova does permit CGMs to be generated with 32-bit precision.

g. The Font List, when present, shall contain no more than four fonts, whose names are drawn from the list of 16 Hershey font names given in Table VI of MIL-D-28003.

Analysis: Most implementations do not include Font List. Of the five that do, three restrict themselves to no more than four names drawn from the approved list of 16 Hershey fonts.

h. The Character Set List must be present and shall contain exactly the two list elements (0,4/2) and (1,4/1).

Analysis: Only two implementations--McDonnell Douglas and System One Software--use the Character Set List. However, neither sets its parameters to the proscribed elements for CALS.

i. The Character Coding Announcer shall be either 0 (BASIC 7-BIT) or 1 (BASIC 8-BIT).

Analysis: All four implementations that use the Character Coding Announcer use one of the two basic values.

4.2.1.3 Picture Descriptor Elements

The SCALING MODE metric parameter is always a floating-point number of precision (9,23).

Analysis: Both of the two implementations that use SCALING MODE metric do so correctly, although an early release of the McDonnell Douglas CGM toolkit wrote the metric scale factor as a fixed-point number.

4.2.1.4 Control Element

a. VDC Integer Precision shall be 16 or 32.

Analysis: No implementation fails to meet this requirement.

b. VDC Real Precision shall be (0,9,23) or (1,16,16).

Analysis: No implementation offering real VDCs fails to meet this requirement.

4.2.1.5 Graphical Primitives

No GDPs may appear, except those authorized by MIL-D-28003. At present, no such GDPs are authorized.

Analysis: Three implementations--ATC, Pansophic D-PICT, and Prime will write GDPs to the CGM. It is not known whether the Computer Associates CGM toolkit will do so.

4.2.1.6 Attribute Elements

a. Line bundle indices shall lie between 1 and 5.

Analysis: Of the four implementations that use bundles, none restrict the range of the index to these limits.

b. Line types shall lie between 1 and 5 or between -11301 and-11308.

Analysis: Ten of the implementations meet this requirement.

c. Marker bundle indices shall lie between 1 and 5.

Analysis: Of the four implementations that use bundles, none restrict the range of the index to these limits.

d. Marker types shall lie between 1 and 5.

Analysis: Six of the implementations meet this requirement.

e. Text bundle indices shall be either 1 or 2.

Analysis: Of the four implementations that use bundles, none restrict the range of the index to these limits.

f. Text font indices shall lie between 1 and 4.

Analysis: Six of the implementations meet this requirement.

g. Character set indices shall be either 1 or 2.

Analysis: Only one implementation--Genigraphics-uses Character Set Index and it does meet the requirement.

h. Alternate character set indices shall be either 1 or 2.

Analysis: None of the implementations use alternate character set index.

i. Fill bundle indices shall lie between 1 and 5.

Analysis: Of the four implementations that use bundles, none restrict the range of the index to these limits.

j. Hatch indices shall lie between 1 and 6 or between -11401 and -11418.

Analysis: Of the 17 implementations that use hatch index, 8 of them meet the requirement.

k. Edge bundle indices shall lie between 1 and 5.

Analysis: None of the implementations use edge bundles.

l. Edge types shall lie between 1 and 5.

Analysis: Only the Zenographics Metafile translator from AutoCAD DXF files to CGM appears not to meet this requirement.

m. For pattern tables, the starting index shall lie between 1 and 8. Each pattern can have no more than 16 rows and 16 columns.

Analysis: Of the five implementations that support pattern tables, none appear to meet this requirement.

n. For color tables, the starting index shall lie between 0 and 255.

Analysis: No implementations are known to violate this requirement.

4.2.1.7 Escape Elements

No ESCAPE elements may appear, except those authorized by MIL-D-28003. At present, three such ESCAPEs are authorized. They have identifiers -301, -302 and -303.

Analysis: No implementations support the authorized ESCAPEs, but ten implementations do permit other private ESCAPEs to appear in the metafile.

4.2.1.8 External Elements

If the MESSAGE element appears, the "action required" flag shall have the value, "no action required."

Analysis: Of the 11 implementations that permit MESSAGE to appear in the CGM, only four appear to restrict the value of the "action required" flag.

4.2.2.1 Additional Linetypes

Eight additional linetypes, with parameter values -11301 through -11308, are authorized for use in CALS basic CGMs.

Analysis: There has not been time for this fairly recently added requirement to be supported in CGM generator products.

4.2.2.2 Additional Hatch Styles

Eighteen additional hatch styles, with parameter values -11401 through -11418, are authorized for use in CALS basic CGMs.

Analysis: There has not been time for this fairly recently added requirement to be supported in CGM generator products.

4.3 Other Requirements

4.3.1 Encoding Format

Only the Binary Encoding shall be used for conforming basic metafiles.

Analysis: In the US, all implementations studied and almost all implementations known can produce at least the Binary Encoding. The workstation-level implementations (Sun, HP, CAl) tend to be able to support all three encodings. In Europe, the first encoding supported has tended to be the Character Encoding.

4.3.2 Physical File Structure

All CGMs shall be organized into 80-octet records.

Analysis: This requirement is rarely met by any implementation. On PC-level products and those developed in C under UNIX usually avoid any logical record structure. Instead, the binary CGM is viewed as a continuous stream of octets. On other workstation-level products (for example, those running under VAX/VMS) prefer to have a logical record size much larger than 80 octets. Record sizes of 256 and 512 are much more prevalent. Only those products with a 'TOP/BASIC' mode appear able to create 80-octet records.

4.3.3 Default Text Precision

Either a TEXT PRECISION 2 (stroke) element is present explicitly at the start of each picture or a METAFILE DEFAULT REPLACEMENTS element setting text precision to 2 is present.

Analysis: Eight of the implementations meet this requirement. Generally, they are the same ones that have a TOP/BASIC mode.

4.3.4 Default Color Table

If color selection mode is indexed and if some color indices are selected (used as attributes) but not set explicitly by a COLOR TABLE element, the intended colors should be as specified in MIL-D-28003, Table VII.

Analysis: Thirteen meet this requirement and eight do not. The other two implementations use direct, rather than indexed color.

4.3.5 Fonts

No font names other than those listed in MIL-D-28003, Table VI, shall be allowed to appear in a FONT LIST ELEMENT. The CGM shall be generated either without any assumptions concerning the font metrics of the selected fonts or assuming only that the metrics match the metrics of the Hershey fonts specified in Table VI.

Analysis: Of the four implementations that use Font List, none meet the exact details of this requirement. However, three of them do cite only Hershey font names, but the names are not constructed exactly in accordance with Table VI.

4.3.6 Metafile Defaults Replacement

This element shall not be partitioned. Instead, multiple instances of this element should be used.

Analysis: Of the four implementations that use Metafile Defaults Replacement (MDR), none are known to partition the MDR element. However, the McDonnell Douglas is known to partition other elements and might partition this element.

4.3.7 Handling of Out-of-range Parameter Values

When a basic conforming generator receives (from any client) a value outside the Basic set, it shall be handled as follows:

- If the index is selecting an attribute (e.g., linetype), then the value shall be mapped by MODULO onto the Basic range;

- If the index is defining an attribute (e.gg a color table entry), then it shall be ignored if outside the Basic range.

Analysis: There is no way to determine whether these requirements are being met by examining the CGM itself. One would have to look at the source code for the CGM generator or else operate the application generating the CGM.

4.3.8 Maximum List Sizes and Lengths

a. Cell arrays may not exceed 1,048,576 elements (one 1024 x 1024 image) in length.

Analysis: Nine implementations can generate a Cell Array element. It is not known whether this restriction is met.

b. Pattern tables may not exceed 2048 elements (eight 16 x 16 patterns) in length.

Analysis: Of the five implementations that use pattern tables, none are willing to limit themselves to this small a number.

c. Color tables may not exceed 256 entries in length.

Analysis: All implementations appear to meet this requirement, although this information was not available for four implementations.

d. No point list array can exceed 1024 (X,Y) pairs.

Analysis: No implementations are known to violate this restriction, although this information was not available for 12 implementations.

e. No data record may exceed 32767 characters.

Analysis: No implementation is known to violate this constraint.

f. No other string parameter may exceed 256 characters.

Analysis: No implementations are known to violate this restriction, although this information was not available for 15 implementations.

5.0 Conformance Testing

It is assumed that the recommendations of section V are accepted so that CGM Certification can be discussed in much greater detail in this section of the report.

5.1 Levels of Testing

Four possible levels of testing are examined:

Level 1: Testing individual instances of a CGM for conformance to the CGM standard, as documented in FIPS PUB 128, ANSI/X3.122, and ISO 8632.

Level 2: Testing individual instances of a CGM for conformance to the CALS CGM Application Profile, as documented in MIL-D-28003.

Level 3: Establishing a Testing Service to verify conformance of

a CGM generator to the CALS CGM Application Profile (AP), as documented in MIL-D-28003.

Level 4: Establishing a Testing Service to verify conformance of a CGM interpreter to the CALS CGM AP, as documented in MIL-D-28003.

Level 1 testing is the only testing that can be performed against the baseline CGM standard, because the standard specifies only the contents of a conforming metafile; the standard does not specify the behavior of conforming CGM generators and interpreters.

The CALS Application Profile for CGM fills the gap left by the baseline CGM standard, by specifying the behavior of conforming basic generators and interpreters. The CALS Application Profile also places additional conformance requirements on CGMs themselves.

It is highly preferable that the testing capabilities be developed in the order given by the level number because a level 3 testing service depends on having a level 2 testing capability and level 2 testing depends on having a level 1 testing capability. Strictly speaking, a level 4 testing service could be established independently of the other three levels of testing; however, having those other testing capabilities available first will make it easier to verify that the CGMs used to test the behavior of CGM interpreters are indeed conforming CGMs.

5.2 Overview of Required Tasks

References 8 and 9 present a global model of the overall Conformance Testing Policy and Procedures that can be applied to Graphics Standards. Tasks can be categorized as either administrative or technical. In the context of Conformance Testing, there are three main participants: the certification body, the testing laboratory, and the client.

For a successful Testing Service to be established, one must design and carry out a series of tasks related to supporting each role expected of the main participants. In the remainder of this section, the general tasks that must be accomplished during the start-up phase are described. Also described is how these general tasks can be particularized to the situation relating to the testing of MIL-D-28003 and roughly estimate the resources required to accomplish each task.

It should be noted that the accomplishment of all the tasks discussed below would constitute an extremely large level of effort by NIST/NCTL, far beyond the scope of present resources.

A significant increase in CALS funding would be needed for NIST/NCTL to hire additional personnel for this effort.

The maintenance phase of Conformance Testing is not addressed in this report. There are on-going requirements to maintain the test methods and oversee the operation of the testing laboratories, but it is beyond the scope of this effort to estimate the costs associated with the long-term maintenance of a CGM Conformance Testing program. Consequently, the resource estimates in the paragraphs marked **Estimated Effort** generally cover only the start-up costs to be incurred within the first 12 months.

The next section elaborates on the tasks associated with the operation of the Testing Laboratory. These are the tasks that the Government is most likely to obtain from a variety of external sources; the other tasks could be accomplished to a large degree using internal resources. **Note that development of the Test Method, especially the creation of the Test Requirements document, the Test Suite and any testing tools, need not be done by a Testing Laboratory itself.** Indeed, because it is desirable that the test suite and testing tools be based on already existing software, it is highly likely that the most cost-effective approach will be to procure these components of the Test Method from third-party sources, not presently affiliated with any Testing Laboratory.

5.2.1 Scope of the Testing Effort

As noted previously, the CALS Application Profile for CGM (MIL-D-28003) makes statements about conforming basic metafiles, conforming basic generators, and conforming basic interpreters. The notion of issuing validation certificates should be applied only to the testing of conforming generators and conforming interpreters (level 3 and level 4 testing), and not to the testing of specific instances of CGMs (level 1 and level 2 testing).

Nevertheless, many tasks associated with the operation of the Certification Body must be accomplished at least in part if one seeks to establish a credible level 1 and level 2 testing program. Consequently, it is highly desirable that one develop and make available to industry an automated testing tool that can check each instance of a purported conforming basic metafile and determine whether it indeed is conforming.

Separate, but related, efforts will be required to establish Conformance Testing Programs for CALS basic conforming generators (level 3 testing) and for CALS basic conforming interpreters (level 4 testing). In particular, although administrative

procedures and processes may be shared, the test methods are quite different.

5.2.2 Initial Tasks Associated With Certification Body Responsibilities

In the task estimates that follow, the following notation is used, **m;n;o;p man-weeks**, to mean that **m** man-weeks are required to implement level 1 testing, an additional **n** man-weeks are required for level 2 testing, an additional **o** man-weeks are required for level 3 testing, and an additional **p** man-weeks are required for level 4 testing.

TCB1. Develop the overall conformance testing program policies and procedures.

Relatively informal policies and procedures are required for level 1 and level 2 testing. It is presumed that the detailed procedures could be abstracted directly from References 8 and 9. The ISO work is especially applicable to the CGM standard. However, testing to MIL-D-28003 requires further detailing of the processes and procedures, because one needs to take into consideration the maintenance and update cycle associated with MIL-D-28003 itself, in addition to any changes in the baseline CGM standard as represented by FIPS PUB 128.

Estimated Effort: 1.5;0.5;2.0;1.0 man-weeks.

TCB2. Approve the design of the test method, when it has been developed.

In fact, two complete test methods must be approved--one for generators and one for interpreters. In addition, the design of the test method for interpreters must be structured to handle implementations that purport to meet only the "draft" conformance level as well as those that aim at meeting the "publication" conformance level. The test method design for generators applies to level 1, 2, and 3 testing.

NIST has recommended that a test method for generators be developed prior to a test method for interpreters.

Estimated Effort: 0.5;0.5;2.0;2.0 man-weeks.

TCB3. Conduct a "trial use" implementation phase, involving a "field test" of the test method.

The provisions of clause 7.2.1, Reference 9, "Acceptance of the Test Suite," can be applied. This clause specifies an "alpha" testing period, where the test method is used by its developers

and the certification body (if desired), and a "beta" testing period, where potential clients test the test method.

"Trial use" of generator and interpreter test methods can be carried out independently of one another.

Estimated Effort: 1;1;2;6 man-weeks for the alpha phase. 1;1;2;4 man-weeks for the beta phase.

TCB4. Approve any revisions to the test method as a result of the "field test."

The certification body and the testing laboratory should work together to decide upon the changes necessitated by the "field test."

Estimated Effort: 1;1;1;1 man-weeks for the alpha phase and 2;2;2;2 man-weeks for the beta phase.

TCB5. Approve the test report form.

The provisions of clause 6.2.2, Reference 9, "Test Specifications," and clause 7.3, "Adoption of a Test Report Format," can be applied.

Estimated Effort: 1;1;1;1 man-weeks.

TCB6. Develop the validation procedures.

As specified in section 4.1B of Reference 8, the validation procedures "define the steps and criteria by which FIPS conformance testing is to be done in order for a validation certificate to be issued." The provisions of clause 7.2.2.4, Reference 9, "On-site Testing Procedures," can be applied.

Estimated Effort: 0;0;1;2 man-weeks. No formal certification is performed with level 1 and level 2 testing.

TCB7. Determine and document the criteria upon which a validation certificate will be issued.

The provisions of clause 8.2, Reference 9, "Organizational Procedures for Certification," and clause 9.2, "Conformance Requirements," can be applied.

Estimated Effort: 0;0;1;3 man-weeks. No formal certification is performed with level 1 and level 2 testing.

TCB8. Design the validation certificate.

Acceptability to the international community should be a high-priority consideration at this stage. One should examine

existing models, both within the graphics community (e.g., GKS) and outside of it (e.g., OSI interchange standards).

Estimated Effort: 0;0;1;1 man-weeks. No formal certification is performed with level 1 and level 2 testing.

TCB9. Establish and document the criteria for accreditation of the testing laboratory.

The provisions of clause 8.1.1.1, Reference 9, "Certification Body and Accreditation Body," and clause 8.1.2, "Test Method Administration," can be applied. Experience with setting up the GKS Testing Service should also be directly applicable.

Estimated Effort: 0;0;4;4 man-weeks. No formal certification is performed with level 1 and level 2 testing so no accreditation of Testing Laboratories is required.

TCB10. Conduct initial proficiency testing of laboratory personnel.

Overseeing "dry runs" by laboratory personnel is the best method of judging their proficiency. Prior to that, personnel should be trained on the procedures to be followed and their understanding of the importance of documentation (Ref. 9, clause 7.2.2.7) and confidentiality (Ref. 9, clause 7.2.2.6). On a more elaborate scale, one could design a written and oral examination that laboratory personnel could be required to pass.

Estimated Effort: 0;0;6;14 man-weeks. No formal certification is performed with level 1 and level 2 testing so no training of Testing Laboratories personnel is required.

TCB11. Verify that the testing laboratory's recordkeeping system is adequate and in accordance with the testing procedures.

The provisions of clause 7.2.2.7, Reference 9, "Documentation," and clause 7.2.2.8, "Archiving of Records," apply. Experience with setting up the GKS Testing Service should also be directly applicable.

Estimated Effort: 0;0;2;2 man-weeks. No formal certification is performed with level 1 and level 2 testing so no checking of the Testing Laboratories' records is required.

TCB12. Develop procedures for resolving disputes. Establish Control Board.

The provisions of clause 7.2.2.1, Reference 9, "Control Board," and clause 7.2.2.2, "Control Board Procedures," apply. Experience with setting up the GKS Control Board should also be directly applicable.

A combined Control Board for CGM (covering both generators and interpreters) is recommended. If the Control Board is mainly concerned with testing to ISO 8632, it is unclear who will interpret issues deriving directly from MIL-D-28003. This could be a very "sticky" matter; consequently, it needs to be discussed very early in the start-up phase and needs to be under constant discussion until a resolution satisfactory to all parties can be obtained.

Estimated Effort: 3;1;2;3 man-weeks, including two or three international meetings to discuss establishment of the CGM Testing Service and the CGM Control Board.

TCB13. Acquire rights to and establish fees for products and services associated with the Conformance Testing program.

Reference 9 (In particular, clauses 7.4, 8.1.1.1, 8.1.1.6, and 8.2) contains guidance regarding the licensing of test method materials and the fees that can be charged for products and services.

Estimated Effort: 2;0.5;1;0.5 man-weeks, including any legal review necessary to approve the language required to implement any "agreements in principle" negotiated between the Government and private developers.

5.2.3 Initial Tasks Associated With Testing Laboratory Operations

No resource estimates shall be given for the tasks listed in this section or in the next section. Only a brief discussion of each task will occur here. Much more detailed discussions and estimates are provided in section 5.3 below.

Remember that these tasks would most likely be developed by third-party suppliers, not Testing Laboratory personnel; the results of the tasks would be used by Testing Laboratory personnel to perform CGM validation and certification.

TTL1. Design the overall test method.

For the generator (levels 1, 2, and 3 testing), the test method will depend on using a reference implementation of a CGM interpreter. A level 3 testing service would require a representative sample of CGMs output by the generator-under-test to be tested for conformance to the conforming basic metafile provisIons of MIL-D-28003.

For the interpreter (level 4 testing), the test method will include the preparation of a set of test CGMs, with associated

hard-copy and documentation. The results of interpreting the CGMs in the Test Set by the interpreter-under-test will be compared visually with the reference hard-copy and conformance established according to documented criteria.

TTL2. Document what parts of the standard are, in fact, to be tested by the test suite. Similarly, document those areas that are not to be tested by the test suite.

The provisions of clause 6.1.1, Reference 9, "Determination of Testing Domain," and clause 6.2.1, "Test Requirements Document," apply.

Most of this work needs to be completed even before the remaining tasks for level 1 testing can be accomplished.

TTL3. Develop the test suite.

The provisions of clause 6.1.2, Reference 9, "Structure of a Test Suite," and clause 6.1.4, "Portability of a Test Suite," apply.

TTL4. Develop any required support tools.

The provisions of clause 6.1.5, Reference 9, "Language Bindings and Encodings," apply.

TTL5. Design the test report, both content and presentation layout.

The provisions of clause 6.2.2, Reference 9, "Test Specifications," and clause 7.3, "Adoption of a Test Report Format," apply.

TTL6. Develop test procedures, including instructions for installing and executing the test suite.

The provisions of clause 6.2.2, Reference 9, "Test Specifications," clause 6.2.3, "Test Suite and Documentation," clause 7.2.2.4, "On-site Testing Procedures," and clause 7.2.2.8, "Checklist," apply.

TTL7. Document how to evaluate the test results for accuracy and completeness.

The provisions of clause 6.2.2, Reference 9, "Test Specifications," and clause 7.2.2.5, "Preparation of the Test Report," apply.

TTL8. Participate in a "field test" of the testing method for public review and comment during a "trial use" period.

The provisions of clause 7.2.1, Reference 9, "Acceptance of the Test Suite," apply.

TTL9. Refine the test method as a result of the "trial use" period.

Collaborate with the Certification Body and/or Accreditation Body to arrive at consensus on what changes need to be made.

TTL10. Develop, negotiate, and approve appropriate legal documents and procedures that protect both the rights of the Government and the test method owner in accordance with the provisions of Reference 8.

In particular, license agreements with the Government, the Certification Body, other Testing Laboratories, and the clients need to be drafted and approved for use.

The provisions of clause 7.4, Reference 9, "Issue of Licenses," apply.

TTL11. Develop change control and maintenance procedures for the test method.

The provisions of clause 6.1.3, Reference 9, "Maintenance of a Test Suite," and clause 7.4, "Maintenance Requirements," apply.

5.2.4 Initial Tasks Associated With Client Responsibilities

These tasks are applicable only to level 3 and level 4 testing, because establishing a formal testing service is not appropriate for level 1 and level 2 testing (the testing of individual CGMs).

TCL1. Specify and document the steps that a client shall follow in order to have a product tested and to obtain a validation certificate.

The provisions of clause 7.2.2.3, Reference 9, "Applying for Testing," apply.

TCL2. Specify and document the statements or information provided by the client when applying for conformance testing leading towards a validation certificate.

In particular, a questionnaire needs to be developed for clients to fill out and submit with their application for testing. The questionnaire will require specifics about the environment on which their software runs as well as about the exact CGM elements, types, and precisions handled by their implementation.

5.3 Detailed Task Descriptions

This section elaborates on the tasks associated with the operation of the Testing Laboratory. Remember, these are the tasks that the Government, represented by the Certification Body, is most likely to obtain from a variety of external sources. Typically, they will **not be performed** by Testing Laboratory personnel.

5.3.1 Task Notation

As noted previously, separate, but related, efforts will be required to establish Conformance Testing Programs for CALS conforming basic generators and for CALS conforming basic interpreters. In particular, although administrative procedures and processes may be shared, the test methods are quite different.

The following sections address the tasks separately--once for testing generators (the Task numbers will have the suffix G) and once for testing interpreters (the Task numbers will have the suffix I). In addition, where the tasks have been subdivided in order to provide more detail, lowercase Roman letters are used as a further suffix.

5.3.2 Testing for Conforming CGM Generators

TTL1G. Design the overall test method.

For the generator, the test method will include the use of a reference implementation of a CGM interpreter. A representative sample of CGMs output by the generator-under-test will be tested for conformance to the **conforming basic metafile** provisions of MIL-D-28003.

Two levels of testing could be performed--one **syntactic** and one **semantic**. For syntactic testing, one wants to develop a process that reads a CGM and automatically produces a Conformance Report. For example, one should, at a minimum:

- verify that each element in the CGM is **well-formed** (that is, encoded correctly and containing the correct number of parameters of the correct data type);
- check for the correct ordering of the elements; and
- verify that constraints on parameter values are observed.

One approach towards testing for semantic correctness involves the use of a full "reference implementation" of a CGM interpreter that produces a graphical display that could be compared with a hardcopy of the picture supplied by the client for the CGM-under-test. Even if a CGM is syntactically correct, it might not contain the appropriate set of CGM elements necessary to portray the intended picture. This stage of testing would require human evaluation and cannot be automated. The quality of the resulting Conformance Report is directly related to the quality of the reference implementation and the skill of the human evaluator.

Another possible approach towards testing for semantic correctness would require clients to specify, in some semi-formal language or on some structured questionnaire, all the elements they intended to place in each of their test CGMs. The semantic analyzer would disassemble the CGM-under-test and compare the elements observed with the client's declaration as to what was intended to be stored in the CGM. This system could be automated, but places a substantial burden on the client. Work supporting this kind of approach is still at the research stage; this approach will not be considered further as being feasible for CGM generator testing.

Estimated Effort: Two man-weeks to decide how much semantic testing--if any--should be performed, to decide what method of testing should be used, and to document in detail the test method selected for generators. This task should be performed once for all three levels of testing generators.

TTL2G. Document what parts of the standard are, in fact, to be tested by the test suite. Similarly, document those areas that are not to be tested by the test suite.

A Test Requirements Document must be developed from two sources: the CGM standard itself (ANSI/X3.122--FIPS PUB 128) and from the CALS AP for CGM (MIL-D-28003). The requirements document will be used:

1. to specify all the tests that must be carried out by the syntactic analyzer to verify the conformance of any particular CGM-under-test, and

2. to specify the number and contents of the collection of CGMs to be provided by a client for testing in order for the client's generator product to be certified as a conforming generator.

In each area, one will have to determine whether the tests should be performed in any particular order and which requirements are more important for testing purposes. This substantial task can be subdivided into smaller tasks to simplify the job of estimating the work involved.

TTL2Ga. Document constraints on syntax, ordering of elements, parameter ranges, etc. needed for correct implementation of a syntax analyzer to verify generator conformance to FIPS PUB 128.

Estimated Effort: Three man-weeks towards level 1 testing.

TTL2Gb. Document constraints on syntax, ordering of elements, parameter ranges, etc. needed for correct implementation of a syntax analyzer to verify generator conformance to MIL-D-28003.

Estimated Effort: 0.5 man-weeks towards level 2 testing, because the results documented in section III(4.0) above can be used as a starting point, although they need to brought up-to-date to reflect the final draft of MIL-D-28003.

TTL2Gc. Document the range of test CGMs needed from the client to verify generator conformance to FIPS PUB 128.

Estimated Effort: Two man-weeks towards level 3 testing.

TTL2Gd. Document the range of test CGMs needed from the client to verify generator conformance to MIL-D-CGM.

Estimated Effort: An additional 1.5 man-weeks towards level 3 testing, because the results documented in section III(4.0) above can be used as a starting point, although they need to brought up-to-date to reflect the final draft of MIL-D-CGM.

TTL3G. Develop the test suite.

Any computer programs written to automatically analyze or interpret CGM files and associated descriptions shall be written in a standard high-level programming language. All machine-dependent, language- dependent, and operating-system-dependent functionality shall be isolated into well-documented modules. Variations among systems shall be parameterized wherever it is feasible to do so.

Reference 9 suggests that the output of each test case shall give the tester the possibility to trace and easily understand the results either by looking at the report file or the source code. Because the test cases are binary encoded CGMs, a human cannot understand their contents by direct examination. Consequently, a utility task that disassembles and prints the contents of a CGM may need to be developed.

Programs shall be written to be small enough to be run on Personal Computer class machines.

TTL3Ga. Develop the syntax analyzer corresponding to the FIPS requirements.

This tool is needed for level 1 testing.

Estimated Effort: Eight man-weeks if one already has a baseline implementation to build upon. The principal effort would involve checking for completeness, implementing the reporting formats required, and testing. Without a baseline implementation, this could take from 26 to 52 man-weeks.

TTL3Gb. Augment the syntax analyzer to additionally check for the constraints specified in MIL-D-28003.

This augmentation of the basic tool is needed for level 2 testing.

Estimated Effort: Three man-weeks once Task TTL3Ga has been completed.

TTL3Gc. Develop a graphical interpreter that can display any CALS conforming basic CGM and that meets at least the Draft Quality interpreter requirements of MIL-D-28003.

This task could be implemented to support individual CGM testing at level 2 or the task could be deferred to testing at level 3.

Estimated Effort: Twelve man-weeks if one already has a baseline implementation to build upon. The principal effort would involve incorporating the Minimum Quality interpreter requirements of MIL-D-28003, implementing the reporting formats required, and testing. An additional twelve man-weeks will be required to upgrade a Draft Quality interpreter to meet the full Publication Quality requirements of MIL-D-28003.

The estimated effort is heavily dependent upon the capabilities of the underlying graphics system used to render the picture. Portability considerations would suggest GKS, but some of the MIL-D-28003 requirements exceed the capabilities of most GKS systems. Consequently, the interpreter might have to do extensive emulation (e.g., of line types, the CALS hatch styles, and fonts).

Without a baseline implementation to build upon, this effort could easily take from 52 to 104 man-weeks.

TTL4G. Develop any required support tools.

The testing personnel and the client personnel need a reliable utility program that will print out, in a convenient formatted fashion, the contents of a given CGM-under-test. This task should be implemented to support individual CGM testing at levels 1 and 2.

Estimated Effort: Eight man-weeks if one already has a baseline reference implementation to build upon. The principal effort would involve improving the documentation to meet Government standards, revising the reporting formats to meet Conformance Testing standards, and testing of the tool.

The estimated effort is heavily dependent upon the initial parsing and reporting capabilities of the baseline reference implementation, if one can be found.

Without a baseline implementation to build upon, this could easily take from 20 to 40 man-weeks.

TTL5G. Design the test report, both content and presentation layout.

Reference 9, clause 6.2.2, suggests that the test report shall provide a summary of tests passed. In the case of failure, a detailed description of errors (parameter values and other relevant information), helpful information for the implementor to determine errors, and a reference to the standard shall be provided. Clause 7.3 further requires that ISO Guide 45 shall be consulted and that the test report format, as a minimum, shall contain the following information:

- description of product under test.
- description of the test environment.
- a summary giving numbers of test cases passed, failed, withdrawn, and total cases.
- full details for each test case which failed.
- a description of any unexpected results.

Test report formats for each of the test suite programs and any support tools will have to be designed.

Estimated Effort: One man-week for the syntactic checker used in level 1 and 2 testing (see tasks TTL3Ga and TTL3Gb) and one man-week for the semantic checker proposed for level 2 or level 3 testing (see task TTL3Gc).

TTL6G. Develop test procedures.

Extensive documentation for both testing personnel and the client is required.

TTL6Ga. Develop, document, and test procedures for installing each of the testing programs and utility tools.

Estimated Effort: 0.5 man-weeks for the syntactic checker used in level 1 and 2 testing (see tasks TTL3Ga and TTL3Gb) and one man-week for the semantic checker proposed for level 2 or level 3 testing (see task TTL3Gc).

TTL6Gb. Develop, document, and test an operator script for each type of test to be performed.

Estimated Effort: One man-week for the semantic checker proposed for level 2 or level 3 testing. No formal scripts are required for level 1 or level 2 syntactic testing.

TTL6Gc. Develop and document maintenance procedures.

Estimated Effort: One man-week for the syntactic checker when used for level 3 testing (see tasks TTL3Ga and TTL3Gb) and one man-week for the semantic checker when used for level 3 testing (see task TTL3Gc). No formal maintenance is required for level 1 or level 2 syntactic testing.

TTL6Gd. Develop and document procedures for on-site modification of the tests.

Estimated Effort: 0.5 man-weeks for the syntactic checker when used for level 3 testing (see tasks TTL3Ga and TTL3Gb) and 0.5 man-weeks for the semantic checker when used for level 3 testing (see task TTL3Gc). No formal on-site modification procedures are required for level 1 or level 2 syntactic testing.

TTL6Ge. Develop and document procedures regarding the retention of hard copy, the keeping of journals, and the verification that the generator-under-test represents the product for which a certification is being requested.

Estimated Effort: One man-week for level 3 testing.

TTL6Gf. Develop checklists of materials to take for on-site testing and checklists of materials to sad after testing is completed.

Estimated Effort: One man-week for level 3 testing.

TTL7G. Document how to evaluate the test results for accuracy and completeness.

According to clause 6.2.2, Reference 9, "Test Specifications," one needs to provide guidelines to judge test results. These guidelines shall be used to decide if a given test has been passed or, in the case of failure, shall allow the testing laboratory to classify errors according to failure.

These are complex issues requiring consensus among the interested parties, which, in this case, include the CALS user and supplier communities, Accredited Standards Committee X3H3, and ISO/IEC JTC1/SC24. Meetings with Validation and Testing Experts will be required before consensus can be arrived at.

Estimated Effort: One man-week will be required to prepare for level 1 and level 2 testing using the syntactic checker only. An additional two to four man-weeks will be required, including participation at an estimated two international meetings and two or three national meetings over a 12 month period, to prepare for using a semantic checker for either level 2 or level 3 testing.

TTL8G. Participate in a "field test" of the testing method for public review and comment during a "trial use" period.

Clause 7.2.1, Reference 9, "Acceptance of the Test Suite," provides for two rounds of testing--an **alpha test period** carried out by the test suite developers and the testing laboratories and a **beta test period** involving the test laboratories and interested third parties (e.g., prospective clients).

TTL8Ga. Participate in an alpha "field test" of the testing method.

During the alpha test period, one would expect to be spending a considerable amount of time running the testing programs against as large a variety of CGMs as one could get one's hands on.

Estimated Effort: One man-week for the syntactic checker used in level 1 and 2 testing (see tasks TTL3Ga and TTL3Gb) and one man-week for the semantic checker proposed for level 2 or level 3 testing (see task TTL3Gc).

TTL8Gb. Participate in a beta "field test" of the testing method.

During the beta test period, one would expect to be answering a lot of questions regarding installation and procedures and also helping prospective clients to debug their implementations.

Estimated Effort: Six man-weeks for the syntactic checker used in level 1 and 2 testing (see tasks TTL3Ga and TTL3Gb) and 12 man-weeks for the semantic checker proposed for level 2 or level 3 testing (see task TTL3Gc).

TTL9G. Refine the test method as a result of the "trial use" period.

TTL9Ga. Collaborate with the Certification Body and/or Accreditation Body to arrive at consensus on what changes need to be made after the alpha test phase.

Estimated Effort: 1.5 man-weeks for the syntactic checker used in level 1 and 2 testing (see tasks TTL3Ga and TTL3Gb) and 1.5 man-weeks for the semantic checker proposed for level 2 or level 3 testing (see task TTL3Gc).

TTL9Gb. Implement the changes resulting from the alpha test phase.

Estimated Effort: 2 man-weeks for the syntactic checker used in level 1 and 2 testing (see tasks TTL3Ga and TTL3Gb) and 2 man-weeks for the semantic checker proposed for level 2 or level 3 testing (see task TTL3Gc).

TTL9Gc. Collaborate with the Certification Body and/or Accreditation Body to arrive at consensus on what changes need to be made after the beta test phase.

Estimated Effort: 0.5 man-weeks for the syntactic checker used in level 1 and 2 testing (see tasks TTL3Ga and TTL3Gb) and 0.5 man-weeks for the semantic checker proposed for level 2 or level 3 testing (see task TTL3Gc).

TTL9Gd. Implement the changes resulting from the beta test phase.

Estimated Effort: 1.5 man-weeks for the syntactic checker used in level 1 and 2 testing (see tasks TTL3Ga and TTL3Gb) and 1.5 man-weeks for the semantic checker proposed for level 2 or level 3 testing (see task TTL3Gc).

TTL10G. Develop, negotiate, and approve appropriate legal documents and procedures that protect both the rights of the Government and the test method owner in accordance with the provisions of Reference 8.

In particular, license agreements with the Government, the Certification Body, other Testing Laboratories, and the clients need to be drafted and approved for use.

According to clause 7.4, Reference 9, "Issue of Licenses," two types of license shall be required:

- a license for clients to use the test suite.
- a license for testing laboratories to enable them to use the test suite and test procedures within a third party test service.

The procedures allow for the charging of a fair and reasonable license fee.

Estimated Effort: For level 1 and level 2 testing using the syntactic checker, 0.5 man-weeks and one meeting to arrive at an "agreement in principle." An additional 0.5 man-weeks of a lawyer's time to draft and agree upon the exact language of the license. For level 2 and level 3 testing using the semantic

checker, an additional 0.5 man-weeks to arrive at an "agreement in principle."

TTL11G. Develop change control and maintenance procedures for the test method.

The provisions of clause 6.1.3, Reference 9, "Maintenance of a Test Suite," specify that the test suite shall be subject to an agreed review procedure with a publicly known timetable. The procedures shall provide formal mechanisms for reporting errors in the test software, withdrawing invalid tests, and logging changes to the test software.

Clause 7.4, "Maintenance Requirements," specifies that change control procedures shall be required to enable changes to be made to the test suite in a controlled manner. The change control procedures shall encompass the test suite, the test procedures, and the test report format.

Estimated Effort: Not required for level 1 and level 2 testing. 1.5 man-weeks for the syntactic checker and 1.5 man-weeks for one of the semantic analyzers, assuming that the general principles adopted for the GKS Testing Service are accepted for CALS MIL-D-28003 testing. If the assumption is correct, the major effort would involve adapting the GKS procedures (which apply to an Application Programmer Interface standard) to CGM (which is a Data Interchange standard).

TCL1G. Specify and document the steps that a client shall follow in order to have a product tested and to obtain a validation certificate.

The detailed provisions of clause 7.2.2.3, Reference 9, "Applying for Testing," provide for:

- the client's being able to purchase the test suite for his own use;
- the client's submitting a scheduling request if he wants to be officially certified;
- the client's reporting the results of a "dry run" prior to formal testing;
- the client's seeking resolution of any queries he might have; and
- the client's scheduling an on-site test.

With appropriate thought regarding procedures, it might be possible to abolish the need for on-site testing in the case of testing CGM generators. It might be possible for the client to

submit his collection of CGMs--and associated documentation--for testing at the laboratory site. This will be possible only if it is not necessary to observe the generator-under-test directly.

Estimated Effort: This task is applicable only to level 3 testing. Two man-weeks, assuming that the general principles adopted for the GKS Testing Service are accepted for CALS MIL-D-28003 testing. If the assumption is correct, the major effort would involve adapting the GKS procedures (which apply to an Application Programmer Interface standard) to CGM (which is a Data Interchange standard).

TCL2G. Specify and document the statements or information provided by the client when applying for conformance testing leading towards a validation certificate.

In particular, a questionnaire needs to be developed for clients to fill out and submit with their application for testing. The questionnaire will require specifics about the environment on which their software runs as well as about the exact CGM elements, types, and precisions produced by their CGM generator.

From this information, the testing laboratory will direct the client to provide a certain number of CGMs, fitting a variety of element and parameter value usage profiles driven by the documented capabilities of the generator-under-test. The success of CGM Generator Testing depends upon the effective completion of this task. The CGMs provided by the client must be representative of all CGMs producible by his generator implementation, because the client's generator is certified only by validating some collection of the CGMs that the generator has produced.

Estimated Effort: This task is applicable only to level 3 testing. This is a non-trivial task and it is driven by the results of Tasks TTL2Ga and TTL2Gb. Three man-weeks should be sufficient if the Test Requirements document is thorough and complete.

5.3.3 Testing for Conforming CGM Interpreters

All the tasks described in this section relate to level 4 testing as defined above in section 5.1.

TTL1I. Design the overall test method.

For the interpreter, the test method will include the preparation of a set of test CGMs (each collection is known as a CGM Test Set) with associated hard-copy and documentation. The results of interpreting the CGMs in the Test Set by the interpreter-under-test will be compared visually with the reference hard-copy and conformance will be established according to documented criteria.

Two levels of testing must be performed--one for CALS Draft Quality interpreters and one for CALS Publication Quality interpreters. Each Test Set should be targeted at verifying the interpreter's handling of a certain class of pictures or certain groups of elements. Test Sets that deliberately include out-of-range parameter values and illegal CGM syntax should also be included.

Estimated Effort: Three man-weeks and one meeting with Certification Body officials to agree upon what method of testing should be used and to document in detail the test method design selected for interpreters.

TTL2I. Document what parts of the standard are, in fact, to be tested by the test suite. Similarly, document those areas that are not to be tested by the test suite.

A Test Requirements Document must be developed from two sources: the CGM standard itself (ANSI/X3.122--FIPS PUB 128) and from the CALS AP for CGM (MIL-D-28003). The requirements document will be used:

1. to specify all the Test Sets that must be developed for inclusion in the test suite,

2. to specify what allowable differences are permitted among realizations of the CGMs in each Test Set for both Draft Quality and Presentation Quality interpreters, and

3. to specify the information about the client's product needed in order to determine which, if any, of the CGMs in any of the Test Sets need not be interpreted by the interpreter-under-test.

In each area, one will have to determine whether the tests should be performed in any particular order and which requirements are more important for testing purposes. This substantial task can be subdivided into smaller tasks to simplify the job of estimating the work involved.

TTL2Ia. Document the number and contents of the CGM Test Sets required to verify an interpreter's ability to read without error CGMs conforming to FIPS PUB 128.

This sub-task focusses on the ability of the interpreter-under-test to handle CGMs conforming to the FIPS correctly from the syntactic point of view.

Estimated Effort: Three man-weeks.

TTL2Ib. Document the number and contents of the CGM Test Sets required to verify an interpreter's ability to read without error CGMs conforming to the provisions of MIL-D-28003.

This sub-task focusses on the ability of the interpreter-under-test to also handle CGMs conforming to the CALS AP correctly from the syntactic point of view.

Estimated Effort: 1.5 man-weeks, because the results documented in Reference 14 can be used as a starting point, although they need to brought up-to-date to reflect the final draft of MIL-D-28003. This estimate assumes the completion of Task TTL2Ia.

TTL2Ic. Document the modifications needed to the CGM Test Sets in order to verify an interpreter's ability to display correctly-at the Draft Quality level-all pictures contained in CGMs conforming to MIL-D-28003.

This sub-task focusses on the ability of the interpreter-under-test to also handle CGMs conforming to MIL-D-28003--Draft Quality level--correctly from the semantic point of view.

Estimated Effort: 1.5 man-weeks, because the results documented in Reference 14 can be used as a starting point, although they need to brought up-to-date to reflect the final draft of MIL-D-28003. This estimate assumes the completion of Task TTL2Ib.

TTL2Id. Document the modifications needed to the CGM Test Sets in order to verify an interpreter's ability to display correctly-at the Publication Quality level-all pictures contained in CGMs conforming to MIL-D-28003.

This sub-task focusses on the ability of the interpreter-under-test to also handle CGMs conforming to MIL-D-28003--Publication Quality level--correctly from the semantic point of view.

Estimated Effort: 1.5 man-weeks, because the results documented in Reference 14 can be used as a starting point, although they need to brought up-to-date to reflect the final draft of MIL-D-28003. This estimate assumes the completion of Task TTL2Ic.

TTL3I. Develop the test suite.

The effort for each subtask assumes that an interactive utility tool (see Task TTL4I below) has been developed and is available for use by test suite developers.

The first four sub-tasks described in the following use the requirements gathered in the corresponding sub-task under TTL2I above. Each of these sub-tasks requires the completion of the predecessor sub-task prior to its commencement.

TTL3Ia. Construct and assemble the CGM Test Sets required to verify an interpreter's ability to read without error CGMs conforming to FIPS PUB 128.

Estimated Effort: Six man-weeks, depending upon the decisions made about the thoroughness of the test method. This estimate assumes that 100 CGMs (at 2.5 man-hours per CGM) will be needed.

TTL3Ib. Construct and assemble the CGM Test Sets required to verify an interpreter's ability to read without error CGMs conforming to the provisions of MIL-D-28003.

Estimated Effort: Three man-weeks, depending upon the decisions made about the thoroughness of the test method. This estimate assumes that 50 more CGMs (at 2.5 man-hours per CGM) will be needed.

TTL3Ic. Implement the modifications needed to the CGM Test Sets in order to verify an interpreter's ability to display correctly-at the Draft Quality level-all pictures contained in CGMs conforming to MIL-D-28003.

Estimated Effort: Three man-weeks, depending upon the decisions made about the thoroughness of the test method. This estimate assumes that 50 more CGMs (at 2.5 man-hours per CGM) will be needed.

TTL3Id. Implement the modifications needed to the CGM Test Sets in order to verify an interpreter's ability to display correctly-at the Publication Quality level-all pictures contained in CGMs conforming to MIL-D-28003.

Estimated Effort: Three man-weeks, depending upon the decisions made about the thoroughness of the test method. This estimate assumes that 50 more CGMs (at 2.5 man-hours per CGM) will be needed.

TTL3Ie. Document the allowable differences for each CGM in each of the CGM Test Sets in order to verify an interpreter's ability to display correctly-at the Draft Quality level-all pictures contained in CGMs conforming to MIL-D-28003.

Estimated Effort: 1.5 man-hours per CGM. This task is complicated by the fact that the differences an evaluator might see are dependent upon the basic capabilities of the interpreter, which are documented by the client--see Task TCL2I.

TTL3If. Document the allowable differences for each CGM in each of the CGM Test Sets in order to verify an interpreter's ability to display correctly-at the Publication Quality level-all pictures contained in CGMs conforming to MIL-D-28003.

Estimated Effort: An additional one man-hour per CGM, assuming the completion of task TTL3Ie.

TTL4I. Develop any required support tools.

In addition to the tools described in Tasks TTL4Ia and TTL4Ib below, the testing personnel and the client personnel also need a reliable utility program that will print out, in a convenient formatted fashion, the contents of a given CGM from the CGM Test Set. This is the same tool needed under task TTL4G, and the estimated effort for its development will not be repeated here, because it is assumed that TTL4G will have been completed before work on Task TTL4I has begun.

TTL4Ia. Develop a tool to generate specific CGMs for the various CGM Test Sets.

A crucial tool is an interactive program that permits the Test Set Developer to build any CGM needed for the Test Set. NIST recommends that this tool be based on a reference implementation of the CGM that can accept Clear Text Encoded CGMs and produce the equivalent Binary Encoded CGM. Extensions to permit illegal and ill-formed CGMs to be created must be built into the tool. Further extensions will be required to permit elements or formats unique to the Binary Encoding to be created from the Clear Text source. During the work on Tasks TTL2I and TTL3I, further extensions and other kinds of requirements might be identified.

Any computer tool programs written to support the Testing Process shall be written in a standard high-level programming language. All machine-dependent, language-dependent, and operating-system-dependent functionality shall be isolated into well-documented modules. Variations among systems shall be parameterized wherever it is feasible to do so.

Programs shall be written to be small enough to be run on Personal Computer class machines.

Estimated Effort: Twelve man-weeks if one already has a baseline reference implementation to build upon and if only minimal ability to generate erroneous CGMs is provided. The principal effort would involve improving the documentation to meet Government standards, implementing the extensions required, and testing of the tool. An additional 8 man-weeks will be required to develop more sophisticated capabilities for generating erroneous CGMs.

This estimated effort is heavily dependent upon the initial parsing and reporting capabilities of the baseline reference implementation, if one can be found, and on the extent of the required extensions.

Without a baseline implementation to build upon, this effort could easily take from 26 to 52 man-weeks.

TTL4Ib. Develop a tool to enter the results of interpreting each file in each CGM Test Set and generate a summary test report.

The display or hardcopy of dozens, if not hundreds or thousands, of CGMs interpreted by the implementation-under-test will need to be evaluated by a human, with the results being entered into a data base for subsequent summarization and reporting. Some kind of access to a DBMS system, via "fill in the form" interactive screens to record the results of each evaluation, will be required to manage the testing process.

Estimated Effort: 8 to 16 man-weeks if one can find a good DBMS system to build the particular tool upon. The principal effort would involve designing the data entry screens, designing the content of the data base, developing the documentation to meet Government standards, implementing the data entry panels and report generator procedures required, and testing of the tool.

The estimated effort is heavily dependent upon the availability of a good DBMS with a versatile report generator. PC-based products like dBASE III and Rbase appear to be good candidates. They could be placed on a 286 or 386 PC laptop computer and used by testing personnel when recording their evaluations. The raw data from each evaluation could then be brought back to the testing laboratory where the summary report would be generated.

Without an off-the-shelf DBMS and report generator to build upon, this effort could take from 26 to 52 man-weeks.

TTL5I. Design the test report, both content and presentation layout.

Reference 9, clause 6.2.2, suggests that the test report shall provide a summary of tests passed. In the case of failure, a detailed description of errors (parameter values and other relevant information), helpful information for the implementor to determine errors, and a reference to the standard shall be provided. Clause 7.3 further requires that ISO Guide 45 shall be consulted and that the test report format, as a minimum, shall contain the following information:

- description of product under test.
- description of the test environment.

- a summary giving numbers of test cases passed, failed, withdrawn, and total cases.

- full details for each test case which failed.

- a description of any unexpected results.

A unified test report format to record the human evaluation comparing the results from the implementation-under-test with the reference results for each of the CGMs in each of the CGM Test Sets will have to be designed.

Estimated Effort: Two man-weeks.

TTL6I. Develop test procedures.

Extensive documentation for both testing personnel and the client is required.

TTL6Ia. Develop, document, and test procedures for installing the utility tools.

Estimated Effort: Two man-weeks.

TTL6Ib. Develop and document an operator script for each CGM in the Test Set.

Estimated Effort: One man-hour for each CGM. This estimate assumes that Tasks TTL3Ie and TTL3If have been completed. Testing time is not included in this task. It is assumed that it is acceptable to overlap testing of the scripts with the other testing conducted during the alpha test phase.

TTL6Ic. Develop and document maintenance procedures.

Estimated Effort: Two man-weeks. It is assumed that a single procedure can be developed that applies to all CGMs in all CGM Test Sets.

TTL6Id. Develop and document procedures for on-site modification of the tests.

Estimated Effort: Two man-weeks. It is assumed that a single procedure can be developed that applies to all CGMs in all CGM Test Sets.

TTL6Ie. Develop and document procedures regarding the retention of hard copy, the keeping of journals, and the verification that the generator-under-test represents the product for which a certification is being requested.

Estimated Effort: Two man-weeks. There are a lot of details to attend to.

TTL6If. Develop checklists of materials to take for on-site testing and checklists of materials to save after testing is completed.

Estimated Effort: One man-week.

TTL7I. Document how to evaluate the test results for accuracy and completeness.

According to clause 6.2.2, Reference 9, "Test Specifications," one needs to provide guidelines to judge test results. These guidelines shall be used to decide if a given test has been passed or, in the case of failure, shall allow the testing laboratory to classify errors according to failure.

The bulk of this effort is already included in Task TTL6I above. However developing general guidelines and principles from which Task TTL6I would derive its specific instructions is a complex matter requiring consensus among the interested parties, which, in this case, include the CALS user and supplier communities, Accredited Standards Committee X3H3, and ISO/IEC JTC1/SC24. Meetings with Validation and Testing Experts will be required before consensus can be arrived at.

Estimated Effort: Three to six man-weeks will be required, including participation at an estimated two international meetings and two or three national meetings over a 12 month period.

TTL8I. Participate in a "field test" of the testing method for public review and comment during a "trial use" period.

Clause 7.2.1, Reference 9, "Acceptance of the Test Suite," provides for two rounds of testing--an **alpha test period** carried out by the test suite developers and the testing laboratories and a **beta test period** involving the test laboratories and interested third parties (e.g., prospective clients).

TTL8Ia. Participate in an alpha "field test" of the testing method.

During the alpha test period, one would expect to be spending all one's time testing the CGMs in all the CGM Test Sets and the associated the operator scripts for each CGM.

Estimated Effort: One man-hour per CGM plus two man-weeks for testing the procedures and the reporting program.

TTL8Ib. Participate in a beta "field test" of the testing method.

During the beta test period, one would expect to be answering a lot of questions regarding procedures and also helping prospective clients to debug their implementations.

Estimated Effort: One person full time over the duration of the beta testing period, say 12 man-weeks.

TTL9I. Refine the test method as a result of the "trial use" period.

TTL9Ia. Collaborate with the Certification Body and/or Accreditation Body to arrive at consensus on what changes need to be made after the alpha test phase.

Estimated Effort: Three man-weeks and two meetings.

TTL9Ib. Implement the changes resulting from the alpha test phase.

Estimated Effort: 1 man-hour per changed CGM and three additional man-weeks for revising the procedures, scripts, reporting program, and instructions.

TTL9Ic. Collaborate with the Certification Body and/or Accreditation Body to arrive at consensus on what changes need to be made after the beta test phase.

Estimated Effort: Two man-weeks and one meeting.

TTL9Id. Implement the changes resulting from the beta test phase.

Estimated Effort: 1 man-hour per changed CGM and two additional man-weeks for revising the procedures, scripts, reporting program, and instructions.

TTL10I. Develop, negotiate, and approve appropriate legal documents and procedures that protect both the rights of the Government and the test method owner in accordance with the provisions of Reference 8.

In particular, license agreements with the Government, the Certification Body, other Testing Laboratories, and the clients need to be drafted and approved for use.

According to clause 7.4, Reference 9, "Issue of Licenses," two types of license shall be required:

- a license for clients to use the test suite.

- a license for testing laboratories to enable them to use the test suite and test procedures within a third party test service.

The procedures allow for the charging of a fair and reasonable license fee.

Each type of license will be required for each program comprising the generator test suite.

Estimated Effort: One man-week and one meeting to arrive at an "agreement in principle" regarding rights to the utility tools and to the CGMs contained in the various CGM Test Sets. An additional man-week of a lawyer's time and one additional meeting, including the lawyer, to draft and agree upon the exact language of the license.

TTL11I. Develop change control and maintenance procedures for the test method.

The provisions of clause 6.1.3, Reference 9, "Maintenance of a Test Suite," specify that the test suite shall be subject to an agreed review procedure with a publicly known timetable. The procedures shall provide formal mechanisms for reporting errors in the test software, withdrawing invalid tests, and logging changes to the test software.

Clause 7.4, "Maintenance Requirements," specifies that change control procedures shall be required to enable changes to be made to the test suite in a controlled manner. The change control procedures shall encompass the test suite, the test procedures, and the test report format.

Estimated Effort: Three man-weeks, assuming that the general principles adopted for the GKS Testing Service are accepted for CGM-D-28003 testing. If the assumption is correct, the major effort would involve adapting the GKS procedures (which apply to an Application Programmer Interface standard and to source code of computer programs) to CGM (which is a Data Interchange standard and represents a file format and content).

TCL1I. Specify and document the steps that a client shall follow in order to have a product tested and to obtain a validation certificate.

The detailed provisions of clause 7.2.2.3, Reference 9, "Applying for Testing," provide for:

- the client's being able to purchase the test suite for his own use;

- the client's submitting a scheduling request if he wants to be officially certified;

- the client's reporting the results of a "dry run" prior to formal testing;

- the client's seeking resolution of any queries he might have; and

- the client's scheduling an on-site test.

With appropriate thought regarding procedures, it might be possible to minimize the length of time required for on-site testing in the case of testing CGM interpreters. If the client interpreter can produce hardcopy, it should be possible for testing personnel to oversee interpretation at the client site and then bring back the hardcopy for evaluation at the laboratory site. This would permit less skilled and/or less experienced personnel to be used for on-site monitoring of the interpretation.

Estimated Effort: Two to four man-weeks, assuming that the general principles adopted for the GKS Testing Service are accepted for CGM-D-28003 testing. If the assumption is correct, the major effort would involve adapting the GKS procedures (which apply to an Application Programmer Interface standard) to CGM (which is a Data Interchange standard).

TCL2I. Specify and document the statements or information provided by the client when applying for conformance testing leading towards a validation certificate.

In particular, a questionnaire needs to be developed for clients to fill out and submit with their application for testing. The questionnaire will require specifics about the environment on which their software runs as well as about the exact CGM elements, types, and precisions handled by their implementation.

Some of the information provided by the client will influence the criteria used to determine whether an interpreter-under-test meets the CALS AP for Draft Quality CGM Interpreters.

Estimated Effort: This is a non-trivial task and it is driven by the results of Tasks TTL2Ia and TTL2Ib. Three man-weeks should be sufficient if the Test Requirements document is thorough and complete.

IV. SUMMARY AND CONCLUSIONS

This section summarizes the major subsections of section III as follows:

- Section 1 below summarizes section III, subsection 3;
- Section 2 below summarizes section III, subsection 4; and
- Section 3 below summarizes section III, subsection 5.

1.0 Summary of Commercial Implementations

The 100 metafiles studied are broadly representative of the full range of uses in which the CGM can be valuable. Only the following CGM elements did not appear in any of the CGMs studied:

VDC REAL PRECISION
RESTRICTED TEXT
GENERALIZED DRAWING PRIMITIVE
CIRCULAR ARC 3 POINT
CIRCULAR ARC 3 POINT CLOSE
Line, Marker, Text, Fill, and Edge BUNDLE INDEX
ALTERNATE CHARACTER SET INDEX
EDGE WIDTH
MESSAGE

These are CGM elements that represent functionality rarely found in today's graphics applications.

However, several other elements are used infrequently in the metafiles surveyed. These include:

METAFILE DEFAULTS REPLACEMENT
FONT LIST
CHARACTER SET LIST
CHARACTER CODING ANNOUNCER
AUXILIARY COLOR
TRANSPARENCY
DISJOINT POLYLINE
APPEND TEXT
POLYGON SET
CELL ARRAY
CIRCULAR ARC CENTER
CIRCULAR ARC CENTER CLOSE
ELLIPSE
ELLIPTICAL ARC
ELLIPTICAL ARC CLOSE
CHARACTER SET INDEX
The Edge Attributes
FILL REFERENCE POINT
PATTERN TABLE
PATTERN SIZE
ASPECT SOURCE FLAGS

Some companies have learned the value of these elements and have incorporated them in their CGMs. However, all worry that if they use these less well-known elements, there will be no products that can interpret their metafiles.

Only six companies use ESCAPE elements and only two--Lotus and System One Software--use APPLICATION DATA elements. It is not known if the information contained in these elements is vital for the proper display of the picture represented by the CGM. In at least one case--Lotus, the information is not required for displaying purposes; rather, the information would be used when reading in the metafile for further manipulation by an editing program.

The small number of companies using METAFILE DEFAULTS REPLACEMENT and FONT LIST indicates that a substantial education process is required in order to improve interoperability of metafiles containing text. A lesser effort will be required to educate the five companies that don't place a COLOR TABLE element in the CGM even when they use indexed color selection mode.

In this task, the variability of commercial CGM generator implementations was studied and the degree of conformance to the CALS Application Profile for CGM (MIL-D-28003) was assessed. The conclusions are that:

- Most companies should be able to upgrade their CGM generator products to conform to MIL-D-28003 without excessive or burdensome additional investment.

- Upgrading of interpreters to meet the CALS Draft Level of conformance should be straightforward but not trivial; upgrading to the Publication Level of conformance will be more costly and time-consuming.

- Any discrepancies between the CGMs produced by products seeking to conform to MIL-D-28003 will be the result of misunderstandings and not deliberate decisions due to difficulty of implementation.

Consequently, NIST recommends that:

1. The testing of CGM generators to MIL-D-28003 be given priority over the testing of CGM interpreters.

2. CALS support and encourage vendor demonstrations like CALS EXPO and NCGA's Integrate'89 to facilitate vendor education and accelerate vendor compliance with MIL-D-28003.

In section VI, seven specific recommendations are provided concerning the certification of CGM implementations.

2.0 Conformance Checklist Summary--Commercial .vs. MIL-D-28003

A summary of how well the implementations studied meet the requirements of MIL-D-28003 is presented in tabular form in the following pages.

Each of the CALS AP requirements is presented as a row and each of the CGM generator implementations is presented as a column. In the cell at the intersection of a row and column is an indicator of conformance to the CALS AP. The symbols used are as follows:

- Y Means the generator meets the CALS requirement.
- N Means the generator does not meet the CALS requirement.
- \- Means the generator does not use the CGM element on which the requirement falls.
- ? Means that not enough information was available for the implementation in question.
- * When attached to a y or n means that the implementation complies with the spirit of the requirement but deviates in some small way.

		3.2.1	3.2.2	3.2.3	3.2.4	3.2.5	3.2.6
		ATC	AUT	SOP	CAI	GEN	GPT
4.2.1.1	No-ops < 32767	Y	Y	Y	Y	Y	Y
4.2.1.2.a1	MD contains product or company name	N	N	Y	Y	Y	Y
4.2.1.2.a2	MD contains string "MIL-D-28003/BASIC-1"	N*	N	N	N	N*	N
4.2.1.2.b	Integer Precision = 16	Y	Y	Y	Y	Y	Y
4.2.1.2.c	Real Precision=(1,16,16) or (0,9,23)	Y	Y	Y	Y	Y	Y
4.2.1.2.d	Index Precision = 16	Y	Y	Y	Y	Y	Y
4.2.1.2.e	Color Precision=8 or 16	Y	Y	Y	Y	Y	Y
4.2.1.2.f	Color Index Prec.=8or16	Y	Y	Y	Y	Y	Y
4.2.1.2.g	Font List <= 4	-	-	-	-	Y	-
4.2.1.2.h	Char. Set List present & contains (0,4/2)(1,4/1)	N -	N -	N -	N -	N -	N -
4.2.1.2.i	CCA = 0 or 1	-	-	-	-	-	-
4.2.1.3	If metric, SCALING MODE parameter is (0,9,23)	-	-	-	Y	-	-
4.2.1.4.a	VDC Int. Prec.= 16 or 32	Y	Y	Y	Y	Y	Y
4.2.1.4.b	VDC Real Prec.=(1,16,16) or (0,9,23)	Y	-	-	-	-	Y
4.2.1.5	No GDPs allowed	N	Y	Y	?	Y	Y
4.2.1.6.a	1<=Line Bundle Index<=5	N	-	-	-	-	-
4.2.1.6.b	1 <= Line Type <=5 or -11301 <= L.T. <= -11308	N	N	N	?	Y	Y

		3.2.1	3.2.2	3.2.3	3.2.4	3.2.5	3.2.6
		ATC	AUT	SOP	CAI	GEN	GPT
4.2.1.6.c	1<=Marker Bun.Index<=5	N	-	-	-	-	-
4.2.1.6.d	1 <= Marker Type <= 5	N	-	N	?	-	-
4.2.1.6.e	1<=Text Bun. Index <=2	N	-	-	-	-	-
4.2.1.6.f	1<=Text Font Index <=4	N	Y	N	Y	N	N
4.2.1.6.g	1<=Char. Set Index <=2	-	-	-	-	Y	-
4.2.1.6.h	1<=Alt.Char.Set Ind<=2	-	-	-	-	-	-
4.2.1.6.i	1<=Fill Bun. Index <=5	N	-	-	?	-	-
4.2.1.6.j	1<= Hatch Index <=6 or -11401<= H.I.<=-11418	N	-	N	?	-	-
4.2.1.6.k	1<=Edge Bun.Index <=5	-	-	-	-	-	-
4.2.1.6.l	1 <= Edge Type <=5	-	N	-	-	-	-
4.2.1.6.m	1<=Pattern Tab.Ind.<=8 Each Pattern wi 16x16	N	-	-	-	-	-
4.2.1.6.n	0<=ColorTableInd.<=255	?	Y	Y	Y	Y	Y
4.2.1.7	Only ESCAPE Els:-301&2	N	-	Y	N	-	-
4.2.1.8	Only MESSAGE w/ Flag = "no action required"	N	-	-	-	-	-
4.3.1	Uses Binary Encoding	Y	Y	Y	Y	Y	Y
4.3.2	80-Octet Records	Y	N	N	?	N	N
4.3.3	Def. text prec STROKE	N	N	N	N	Y	N
4.3.4	Don't use color indices unless set/follow CALS default table	Y	Y	N	N	Y	Y

		3.2.1	3.2.2	3.2.3	3.2.4	3.2.5	3.2.6
		ATC	AUT	SOP	CAI	GEN	GPT
4.3.5	Only Hershey Font Names in Font List Element	-	-	-	-	N	-
4.3.6	MDR shall not be part'd	-	-	-	Y	-	-
4.3.8.a	Cell arrays<=1,048,576	?	-	?	?	-	-
4.3.8.b	Pattern Tables <=2048	N	-	-	-	-	-
4.3.8.c	Color Tables <= 256 entries	?	Y	Y	Y	Y	Y
4.3.8.d	Point arrays <= 1024 pairs	Y	Y	?	?	Y	?
4.3.8.e	Data records <= 32767	Y	-	Y	Y	Y	Y
4.3.8.f	Strings <= 256	Y	?	?	?	Y	?

		3.2.7	3.2.8	3.2.9	3.2.10	3.2.11	3.2.12
		GSS	HPS	IBM	LOT	MCD	MDT
4.2.1.1	No-ops < 32767	Y	Y	Y	Y	Y	Y
4.2.1.2.a1	MD contains product or company name	Y	Y	Y	Y	Y	Y
4.2.1.2.a2	MD contains string "MIL-D-28003/BASIC-1"	N	N*	N	N	N*	N
4.2.1.2.b	Integer Precision = 16	Y	Y	Y	Y	-	Y
4.2.1.2.c	Real Precision=(1,16,16) or (0,9,23)	Y	Y	Y	Y	Y	Y
4.2.1.2.d	Index Precision = 16	Y	Y	Y	Y	-	Y
4.2.1.2.e	Color Precision=8 or 16	Y	Y	Y	-	Y	Y
4.2.1.2.f	Color Index Prec.=8or16	Y	Y	Y	Y	Y	Y
4.2.1.2.g	Font List <= 4	-	-	-	-	Y	N
4.2.1.2.h	Char. Set List present & contains (0,4/2)(1,4/1)	N -	N -	N -	N -	Y N	N -
4.2.1.2.i	CCA = 0 or 1	-	Y	-	-	-	-
4.2.1.3	If metric, SCALING MODE parameter is (0,9,23)	-	-	-	-	Y	-
4.2.1.4.a	VDC Int. Prec.= 16 or 32	Y	Y	Y	Y	Y	Y
4.2.1.4.b	VDC Real Prec.=(1,16,16) or (0,9,23)	-	Y	-	-	Y	Y
4.2.1.5	No GDPs allowed	Y	Y	Y	Y	Y	Y
4.2.1.6.a	1<=Line Bundle Index<=5	-	-	-	-	-	-
4.2.1.6.b	1 <= Line Type <=5 or -11301 <= L.T. <= -11308	N	N	N	N	Y	Y

		3.2.7	3.2.8	3.2.9	3.2.10	3.2.11	3.2.12
		GSS	HPS	IBM	LOT	MCD	MDT
4.2.1.6.c	1<=Marker Bun.Index<=5	-	-	-	-	-	-
4.2.1.6.d	1 <= Marker Type <= 5	N	N	N	N	Y	Y
4.2.1.6.e	1<=Text Bun. Index <=2	-	-	-	-	-	-
4.2.1.6.f	1<=Text Font Index <=4	N	N	N	N	Y	Y
4.2.1.6.g	1<=Char. Set Index <=2	-	-	-	-	-	-
4.2.1.6.h	1<=Alt.Char.Set Ind<=2	-	-	-	-	-	-
4.2.1.6.i	1<=Fill Bun. Index <=5	-	-	-	-	-	-
4.2.1.6.j	1<= Hatch Index <=6 or -11401<= H.I.<=-11418	N	-	N	N	Y	Y
4.2.1.6.k	1<=Edge Bun.Index <=5	-	-	-	-	-	-
4.2.1.6.l	1 <= Edge Type <=5	-	-	-	-	-	-
4.2.1.6.m	1<=Pattern Tab.Ind.<=8 Each Pattern wi 16x16	-	-	-	-	-	-
4.2.1.6.n	0<=ColorTableInd.<=255	Y	-	Y	-	Y	Y
4.2.1.7	Only ESCAPE Els:-301&2	N	N	N	N	Y	Y
4.2.1.8	Only MESSAGE w/ Flag = "no action required"	N	N	-	-	-	-
4.3.1	Uses Binary Encoding	Y	Y	Y	Y	Y	Y
4.3.2	80-Octet Records	N	Y	N	N	Y	?
4.3.3	Def. text prec STROKE	N	Y	N	N	Y	Y
4.3.4	Don't use color indices unless set/follow CALS default table	N	-	N	N	Y	N

		3.2.7	3.2.8	3.2.9	3.2.10	3.2.11	3.2.12
		GSS	HPS	IBM	LOT	MCD	MDT
4.3.5	Only Hershey Font Names in Font List Element	-	-	-	-	Y*	Y*
4.3.6	MDR shall not be part'd	-	Y	-	-	?	-
4.3.8.a	Cell arrays<=1,048,576	?	-	?	-	-	-
4.3.8.b	Pattern Tables <=2048	-	-	-	-	-	-
4.3.8.c	Color Tables <= 256 entries	Y	-	Y	-	Y	-
4.3.8.d	Point arrays <= 1024 pairs	?	Y	?	?	?	?
4.3.8.e	Data records <= 32767	Y	Y	Y	Y	Y	Y
4.3.8.f	Strings <= 256	?	Y	?	?	?	?

		3.2.13	3.2.14	3.2.15	3.2.16	3.2.17	3.2.18
		NOV	PAN	OPT	PVI	PRM	SPB
4.2.1.1	No-ops < 32767	Y	Y	Y	Y	Y	Y
4.2.1.2.a1	MD contains product or company name	Y	Y	N	Y	Y	Y
4.2.1.2.a2	MD contains string "MIL-D-28003/BASIC-1"	N	N	N*	N*	N	N
4.2.1.2.b	Integer Precision = 16	Y*	Y	Y	Y	Y	Y
4.2.1.2.c	Real Precision=(1,16,16) or (0,9,23)	Y	Y	Y	Y	Y	Y
4.2.1.2.d	Index Precision = 16	Y*	Y	Y	Y	Y	Y
4.2.1.2.e	Color Precision=8 or 16	Y*	Y	Y	Y	Y	Y
4.2.1.2.f	Color Index Prec.=8or16	Y*	-	Y	Y	Y	Y
4.2.1.2.g	Font List <= 4	N	-	-	-	-	-
4.2.1.2.h	Char. Set List present & contains (0,4/2)(1,4/1)	N -	N -	N -	N -	N -	N -
4.2.1.2.i	CCA = 0 or 1	-	-	-	-	-	-
4.2.1.3	If metric, SCALING MODE parameter is (0,9,23)	-	-	-	-	-	-
4.2.1.4.a	VDC Int. Prec.= 16 or 32	Y	Y	Y	Y	Y	Y
4.2.1.4.b	VDC Real Prec.=(1,16,16) or (0,9,23)	Y	-	Y	Y	Y	-
4.2.1.5	No GDPs allowed	Y	Y	N	Y	N	Y
4.2.1.6.a	1<=Line Bundle Index<=5	N	-	N	-	N	-
4.2.1.6.b	1 <= Line Type <=5 or -11301 <= L.T. <= -11308	Y	Y	N	Y	N	N

		3.2.13 NOV	3.2.14 PAN	3.2.15 OPT	3.2.16 PVI	3.2.17 PRM	3.2.18 SPB
4.2.1.6.c	1<=Marker Bun.Index<=5	N	-	N	-	N	-
4.2.1.6.d	1 <= Marker Type <= 5	Y	-	N	Y	N	N
4.2.1.6.e	1<=Text Bun. Index <=2	N	-	N	-	N	-
4.2.1.6.f	1<=Text Font Index <=4	N	-	N	?	N	N
4.2.1.6.g	1<=Char. Set Index <=2	-	-	-	-	-	-
4.2.1.6.h	1<=Alt.Char.Set Ind<=2	-	-	-	-	-	-
4.2.1.6.i	1<=Fill Bun. Index <=5	N	-	N	-	N	-
4.2.1.6.j	1<= Hatch Index <=6 or -11401<= H.I.<=-11418	Y	-	N	Y	N	N
4.2.1.6.k	1<=Edge Bun.Index <=5	-	-	-	-	-	-
4.2.1.6.l	1 <= Edge Type <=5	-	Y	-	Y	-	-
4.2.1.6.m	1<=Pattern Tab.Ind.<=8 Each Pattern wi 16x16	?	-	N	-	N	-
4.2.1.6.n	0<=ColorTableInd.<=255	Y	-	?	Y	?	Y
4.2.1.7	Only ESCAPE Els:-301&2	Y	Y	N	Y	N	?
4.2.1.8	Only MESSAGE w/ Flag = "no action required"	N	Y	N	Y	N	Y
4.3.1	Uses Binary Encoding	Y	Y	Y	Y	Y	Y
4.3.2	80-Octet Records	?	N	Y	?	Y	N
4.3.3	Def. text prec STROKE	Y	N	N	Y	N	N
4.3.4	Don't use color indices unless set/follow CALS default table	N	-	N	Y	Y	Y

		3.2.13	3.2.14	3.2.15	3.2.16	3.2.17	3.2.18
		NOV	PAN	OPT	PVI	PRM	SPB
4.3.5	Only Hershey Font Names in Font List Element	Y*	-	-	-	-	-
4.3.6	MDR shall not be part'd	-	-	-	-	-	-
4.3.8.a	Cell arrays<=1,048,576	?	-	?	-	-	-
4.3.8.b	Pattern Tables <=2048	?	-	N	-	N	-
4.3.8.c	Color Tables <= 256 entries	Y	-	?	Y	?	Y
4.3.8.d	Point arrays <= 1024 pairs	?	?	Y	Y	Y	Y
4.3.8.e	Data records <= 32767	Y	Y	Y	Y	Y	Y
4.3.8.f	Strings <= 256	?	?	Y	Y	Y	Y

		3.2.19	3.2.20	3.2.21	3.2.22	3.2.23	
		SUN	SOS	TKN	WAS	ZEN	
4.2.1.1	No-ops < 32767	Y	Y	Y	Y	Y	
4.2.1.2.a1	MD contains product or company name	Y	Y	Y	Y	N	
4.2.1.2.a2	MD contains string "MIL-D-28003/BASIC-1"	N*	N	N	N	N	
4.2.1.2.b	Integer Precision = 16	Y	Y	Y	Y	Y	
4.2.1.2.c	Real Precision=(1,16,16) or (0,9,23)	Y	Y	Y	Y	Y	
4.2.1.2.d	Index Precision = 16	Y*	Y	Y	Y	Y	
4.2.1.2.e	Color Precision=8 or 16	Y*	Y	Y	Y	Y	
4.2.1.2.f	Color Index Prec.=8or16	Y	Y	Y	Y	Y	
4.2.1.2.g	Font List <= 4	-	Y	-	-	-	
4.2.1.2.h	Char. Set List present & contains (0,4/2)(1,4/1)	N -	Y N	N -	N -	N -	
4.2.1.2.i	CCA = 0 or 1	Y	Y	Y	-	-	
4.2.1.3	If metric, SCALING MODE parameter is (0,9,23)	-	-	-	-	-	
4.2.1.4.a	VDC Int. Prec.= 16 or 32	Y	Y	Y	Y	Y	
4.2.1.4.b	VDC Real Prec.=(1,16,16) or (0,9,23)	-	-	-	-	-	
4.2.1.5	No GDPs allowed	Y	Y	Y	Y	Y	
4.2.1.6.a	1<=Line Bundle Index<=5	-	-	-	-	-	
4.2.1.6.b	1 <= Line Type <=5 or -11301 <= L.T. <= -11308	N	Y	N	Y	Y	

		3.2.19	3.2.20	3.2.21	3.2.22	3.2.23	
		SUN	SOS	TKN	WAS	ZEN	
4.2.1.6.c	1<=Marker Bun.Index<=5	-	-	-	-	-	
4.2.1.6.d	1 <= Marker Type <= 5	N	Y	N	Y	-	
4.2.1.6.e	1<=Text Bun. Index <=2	-	-	-	-	-	
4.2.1.6.f	1<=Text Font Index <=4	Y	Y	N	N	N	
4.2.1.6.g	1<=Char. Set Index <=2	-	-	-	-	-	
4.2.1.6.h	1<=Alt.Char.Set Ind<=2	-	-	-	-	-	
4.2.1.6.i	1<=Fill Bun. Index <=5	-	-	-	-	-	
4.2.1.6.j	1<= Hatch Index <=6 or -11401<= H.I.<=-11418	?	Y	N	Y	-	
4.2.1.6.k	1<=Edge Bun.Index <=5	-	-	-	-	-	
4.2.1.6.l	1 <= Edge Type <=5	-	Y	-	Y	Y	
4.2.1.6.m	1<=Pattern Tab.Ind.<=8 Each Pattern wi 16x16	N	-	-	-	-	
4.2.1.6.n	0<=ColorTableInd.<=255	Y	Y	Y	Y	Y	
4.2.1.7	Only ESCAPE Els:-301&2	Y	N	N	Y	Y	
4.2.1.8	Only MESSAGE w/ Flag = "no action required"	?	Y	-	-	-	
4.3.1	Uses Binary Encoding	Y	Y	Y	Y	Y	
4.3.2	80-Octet Records	?	N	N	N	N	
4.3.3	Def. text prec STROKE	Y	Y	N	N	N	
4.3.4	Don't use color indices unless set/follow CALS default table	Y	Y	Y	Y	Y	

		3.2.19	3.2.20	3.2.21	3.2.22	3.2.23	
		SUN	SOS	TKN	WAS	ZEN	
4.3.5	Only Hershey Font Names in Font List Element	-	-	-	-	-	
4.3.6	MDR shall not be part'd	Y	-	-	-	-	
4.3.8.a	Cell arrays<=1,048,576	?	-	?	-	-	
4.3.8.b	Pattern Tables <=2048	N	-	-	-	-	
4.3.8.c	Color Tables <= 256 entries	Y	Y	Y	?	Y	
4.3.8.d	Point arrays <= 1024 pairs	Y	?	?	?	Y	
4.3.8.e	Data records <= 32767	Y	Y	Y	Y	Y	
4.3.8.f	Strings <= 256	Y	?	?	?	?	

3.0 Summary of Conformance Testing Tasks

3.1 Development Phases

There is a logical progression that one can follow for the introduction of CGM testing services. This approach is cost-effective and is outlined in the following:

Phase 1 Develop, license, and make available for purchase a CGM syntax analyzer (as described in task TTL3Ga) that checks instances of CGMs for conformance to FIPS PUB 128 along with the utility program that prints out the contents of a CGM-under-test (as described in task TTL4G).

This Test Suite would be used by suppliers to verify that the CGMs they produce indeed conform to the FIPS and by users to verify that the CGMs they are receiving conform to the FIPS. No formal testing service would be established at this time.

Phase 2 Develop, license, and make available for purchase an augmentation to the CGM syntax analyzer (as described in task TTL3Gb) that checks instances of CGMs for conformance to MIL-D-28003.

This augmented Test Suite would be used by CALS suppliers to verify that the CGMs they produce indeed conform to MIL-D-28003 and by CALS users to verify that the CGMs they are receiving conform to MIL-D-28003. No formal testing service would be established at this time.

Phase 3 Develop, license, and make available for purchase a CGM graphical interpreter (as described in task TTL3Gc) that can display any CALS basic conforming CGM and that meets at least the Draft Quality interpreter requirements of MIL-D-28003.

The resulting augmented Test Suite would be used by suppliers and users alike to assist in determining **semantic** correctness to supplement the syntactic correctness report provided by Phase 2. Still, no formal testing service would be established at this time.

Phase 4 Establish a testing service to validate and certify CALS CGM generators for syntactic correctness.

This phase involves performing the requirements analysis described in tasks TTL2Gc and TTL2Gd and all

the other tasks listed in sections 3 and 4 above that relate to establishing a formal testing service for CGM generators.

Phase 5 Augment the generator testing service to validate and certify CALS CGM generators for both syntactic correctness and semantic correctness.

The principal additional efforts over Phase 3 and Phase 4 involve: (a) documenting the procedures and criteria associated with the human inspection process implied by checking for semantic correctness; and (b) upgrading the reference implementation CGM interpreter used in Phase 3 to function at the Publication Quality level specified in MIL-D-28003.

Phase 6 Establish a testing service to validate and certify CALS CGM interpreters. Interpreters will be checked for their ability to read CALS conforming basic CGMs and display them at the Draft Quality interpretation level. Very little emphasis will be placed on testing for reaction to out-of-range values and proper handling of non-conforming CGMs.

This phase involves performing most of the tasks specified in section 4.3 above. The effort is based on a Test Suite consisting of 200 CGMs.

Phase 7 Augment the Phase 6 testing service for CALS CGM interpreters. Interpreters will be checked for their ability to read CALS conforming basic CGMs and display them at the Publication Quality interpretation level. As with Phase 6, very little emphasis will be placed on testing for reaction to out-of-range values and proper handling of non-conforming CGMs.

The additional work involves modifying the operator scripts and certification and validation criteria to reflect the requirements of Publication Quality interpretation. This effort assumes that an additional 50 CGMs will be added to the Test Suite.

Phase 8 Augment the Phase 6 or Phase 7 testing service for CALS CGM interpreters. Substantial emphasis will be placed on the interpreter's response to out-of-range values and degenerate primitives. The proper handling of non-conforming CGMs will also be verified.

The additional work involves: (a) upgrading the CGM creation tool to create erroneous CGMs; (b) creating lots of new operator scripts; and (c) augmenting certification and validation criteria to reflect the

requirements of MIL-D-28003. This effort assumes that an additional 200 CGMs will be added to the Test Suite.

3.2 Summary Tables

Table 3-1 lists all the tasks (along with the Task Identifier) associated with the responsibilities of the Certification Body already described in section III, subsection 5.2.2. Table 3-2 lists all the tasks (along with the Task Identifier) associated with the operation of a CGM generator testing service by a Testing Laboratory and the two tasks associated with the responsibilities of the Client. These tasks have been detailed in section III, subsection 5.3.2. Table 3-3 lists all the tasks (along with the Task Identifier) associated with the operation of a CGM interpreter testing service by a Testing Laboratory and the two tasks associated with the responsibilities of the Client. These tasks have been detailed in section III, subsection 5.3.3.

Table 3-4 summarizes the Certification Body man-day estimates for all eight phases. Table 3-5 summarizes the man-day estimates for Phases 1 through 5, related to successively more complete CGM generator testing, while Table 3-6 summarizes the man-day estimates for Phases 6 through 8, related to successively more complete CGM interpreter testing.

Generally speaking, the tasks listed in Table 3-4 represent tasks that would have to be performed by the Certification Body or delegated to some neutral, governmental or quasi-governmental body and performed by in-house personnel. The tasks listed in Tables 3-5 and 3-6 represent tasks that would be expected to be contracted out to a third-party supplier, often referred to as the test method developer. Once the developer has finished his work and the legal terms and conditions negotiated for use ofthe finished products, the results of the tasks would be made available for licensing to the general supplier and user community and would be utilized by Testing Laboratory personnel in performance of their duties as part of any formal Testing Service established by the Certification Body.

Table 3-1. Tasks Associated with the Certification Body

Task ID	Task Description
TCB1	Develop overall policies and procedures.
TCB2	Approve the design of the test method.
TCB3	Conduct "alpha" and "beta" field tests.
TCB4	Approve revisions to test method resulting from field tests.
TCB5	Approve test report forms.
TCB6	Develop validation procedures.
TCB7	Determine validation criteria.
TCB8	Design validation certificate.
TCB9	Establish Testing Laboratory (TL) accreditation criteria.
TCB10	Conduct initial proficiency testing of TL personnel.
TCB11	Verify TL's recordkeeping system.
TCB12	Develop appeal procedures. Establish control board.
TCB13	License testing tools. Establish fees.

Table 3-2. Tasks Associated with Testing Laboratory Operation and Client Responsibilities with respect to CGM Generator Testing

Task ID	Task Description
TTL1G	Design the test method.
TTL2G	Develop test requirements document:
TTL2Ga	Document requirements for conformance to FIPS PUB 128.
TTL2Gb	Document requirements for conformance to MIL-D-28003.
TTL2Gc	Document range of test CGMs needed for FIPS conformance.
TTL2Gd	Document range of test CGMs needed for MIL-D-28003 conformance.
TTL3G	Develop test suite:
TTL3Ga	Develop FIPS syntax analyzer.
TTL3Gb	Develop MIL-D-28003 syntax analyzer.
TTL3Gc	Develop reference CGM interpreter.
TTL4G	Develop utility to print the content of Binary CGMs.
TTL5G	Design the test reports.
TTL6G	Develop test procedures:
TTL6Ga	For test programs and utility tool.
TTL6Gb	Operator scripts for each test.
TTL6Gc	Maintenance procedures.
TTL6Gd	On-site modifications.
TTL6Ge	On-site recordkeeping requirements.
TTL6Gf	Checklists of materials.
TTL7G	Document how to evaluate test results.
TTL8G	Participate in field tests:
TTL8Ga	Alpha field test.
TTL8Gb	Beta field test.
TTL9G	Refine test method as result of field tests:
TTL9Ga	Design changes after alpha results known.
TTL9Gb	Implement alpha changes.
TTL9Gc	Design changes after beta results known.
TTL9Gd	Implement beta changes.
TTL10G	Negotiate license terms and conditions.
TTL11G	Develop change control procedures for the test method.
TCL1G	Specify steps to be followed by client.
TCL2G	Specify information needed from client.

Table 3-3. Tasks Associated with Testing Laboratory Operation and Client Responsibilities with respect to CGM Interpreter Testing

Task ID	Task Description
TTL1I	Design the test method.
TTL2I	Develop test requirements document:
TTL2Ia	Document requirements for conformance to FIPS PUB 128.
TTL2Ib	Document requirements for conformance to MIL-D-28003.
TTL2Ic	Document requirements for Draft Quality interpretation.
TTL2Id	Document requirements for Publication Quality interpretation.
TTL3I	Develop test suite:
TTL3Ia	Construct CGMs to test FIPS PUB 128 syntax conformance.
TTL3Ib	Construct CGMs to test MIL-D-28003 syntax conformance.
TTL3Ic	Construct CGMs to test Draft Quality conformance.
TTL3Id	Construct CGMs to test Publication Quality conformance.
TTL3Ie	Document allowable differences for Draft Quality CGMs.
TTL3If	Document allowable differences for Publication Quality CGMs.
TTL4I	Develop support tools:
TTL4Ia	Develop CGM generation utility.
TTL4Ib	Develop Test Result recording utility based on a DBMS.
TTL5I	Design the test reports.
TTL6I	Develop test procedures:
TTL6Ia	For the utility programs.
TTL6Ib	Operator scripts for CGM in the Test Set.
TTL6Ic	Maintenance procedures.
TTL6Id	On-site modifications.
TTL6Ie	On-site recordkeeping requirements.
TTL6If	Checklists of materials.
TTL7I	Document how to evaluate test results.
TTL8I	Participate in field tests:
TTL8Ia	Alpha field test.
TTL8Ib	Beta field test.

Table 3-3. Tasks Associated with Testing Laboratory Operation and Client Responsibilities with respect to CGM Interpreter Testing (Continued)

Task ID	Task Description
TTL9I	Refine test method as result of field tests:
TTL9Ia	Design changes to Test Set after alpha results known.
TTL9Ib	Implement alpha changes.
TTL9Ic	Design changes to Test Set after beta results known.
TTL9Id	Implement beta changes.
TTL10I	Negotiate license terms and conditions.
TTL11I	Develop change control procedures for the test method.
TCL1I	Specify steps to be followed by client.
TCL2I	Specify information needed from client.

Table 3-4. Tasks Associated with the Certification Body
(all estimates in man-weeks)

Task Id	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5	Phase 6
TCB1	1.5	0.5	---	2.0	---	1.0
TCB2	0.5	0.5	---	2.0	---	2.0
TCB3	2.0	2.0	2.0	4.0	4.0	6.0
TCB4	3.0	3.0	3.0	3.0	3.0	3.0
TCB5	1.0	1.0	1.0	1.0	1.0	1.0
TCB6	---	---	---	1.0	---	2.0
TCB7	---	---	---	1.0	---	3.0
TCB8	---	---	---	1.0	---	1.0
TCB9	---	---	---	4.0	---	4.0
TCB10	---	---	---	6.0	---	14.0
TCB11	---	---	---	2.0	---	2.0
TCB12	3.0	1.0	---	2.0	---	3.0
TCB13	2.0	0.5	1.0	1.0	---	0.5
Totals	**13.0**	**8.5**	**7.0**	**30.0**	**8.0**	**42.5**

Table 3-5. Tasks Associated with Testing Laboratory Operation and Client Responsibilities with Respect to CGM Generator Testing
(all estimates in man-weeks)

Task ID	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5
TTL1G	2.0	---	---	---	---
TTL2Ga	3.0	---	---	---	---
TTL2Gb	---	0.5	---	---	---
TTL2Gc	---	---	---	2.0	---
TTL2Gd	---	---	---	1.5	---
TTL3Ga	8.0*	---	---	---	---
TTL3Gb	---	3.0*	---	---	---
TTL3Gc	---	---	12.0*	---	12.0*
TTL4G	8.0*	---	---	---	---
TTL5G	1.0	---	1.0	---	---
TTL6Ga	0.5	---	1.0	---	0.5
TTL6Gb	---	---	---	1.0	0.5
TTL6Gc	---	---	---	2.0	0.5
TTL6Gd	---	---	---	1.0	0.5
TTL6Ge	---	---	---	1.0	0.5
TTL6Gf	---	---	---	1.0	0.5
TTL7G	1.0	---	2.0	2.0	1.0
TTL8Ga	1.0	1.0	2.0	2.0	2.0
TTL8Gb	6.0	6.0	12.0	10.0	6.0
TTL9Ga	1.5	1.5	1.5	1.5	1.5
TTL9Gb	2.0	2.0	2.0	2.0	2.0
TTL9Gc	0.5	0.5	0.5	0.5	0.5
TTL9Gd	1.5	1.5	1.5	1.5	1.5
TTL10G	1.0	1.0	0.5	0.5	0.5
TTL11G	---	---	---	3.0	1.5
TCL1G	---	---	---	2.0	2.0
TCL2G	---	---	---	3.0	2.0
Totals	37.0	17.0	36.0	37.5	36.5

* Assumes the availability of a baseline implementation.

Table 3-6. Tasks Associated with Testing Laboratory Operation and Client Responsibilities with Respect to CGM Interpreter Testing (all estimates in man-weeks)

Task ID	Phase 6	Phase 7	Phase 8
TTL1I	3.0	---	---
TTL2Ia	3.0	---	---
TTL2Ib	1.5	---	---
TTL2Ic	1.5	---	---
TTL2Id	---	1.5	---
TTL3Ia	6.0	---	---
TTL3Ib	3.0	---	---
TTL3Ic	3.0	---	---
TTL3Id	---	3.0	---
TTL3Ie	7.5	2.0	7.5
TTL3If	---	6.5	---
TTL4Ia	12.0*	---	8.0*
TTL4Ib	8.0*	4.0*	4.0*
TTL5I	1.0	---	1.0
TTL6Ia	2.0	---	---
TTL6Ib	5.0	1.5	5.0
TTL6Ic	2.0	---	---
TTL6Id	2.0	---	---
TTL6Ie	1.0	---	1.0
TTL6If	1.0	---	1.0
TTL7I	3.0	---	3.0
TTL8Ia	7.0	8.5	13.0
TTL8Ib	12.0	12.0	12.0
TTL9Ia	3.0	1.5	3.0
TTL9Ib	3.5	4.0	6.0
TTL9Ic	2.0	1.0	2.0
TT19Id	1.5	2.0	3.0
TTL10I	2.0	---	---
TTL11I	3.0	---	---
TCL1I	2.0	---	2.0
TCL2I	2.0	---	1.0
Totals	108.5	47.5	72.5

* Assumes the availability of a baseline implementation

V. ASSESSMENT OF IMPACT ON CGM TESTING

1.0 Impact on the CALS Program

Implementation of a full CGM (MIL-D-28003) testing program is vital if CALS is to reap the benefits of interoperability potentially offered by conforming CGM interpreters and generators.

More than most information processing standards and typical of data interchange standards (like IGES), the CGM standard offers implementors a wide range of options. Files can be tested for conformance to the standard, but interoperability is not guaranteed even when conforming CGMs are used for interchange. This situation results when the options selected by the generating program do not match the options supported by all the target interpreting programs.

Until and unless CALS supports the development of a formal testing program for MIL-D-28003, the full potential of CGM as a compact, efficient, and powerful data interchange format for geometric graphic information (as contained in technical drawings, for example) cannot be realized. In the absence of testing to the CALS Application Profile, commercial companies will use only a minimal subset of CGM (just as they have done for IGES). Consequently, the resulting files will be less compact and more costly to transmit, store, and process. In addition, they might poorly represent the intended picture as, for example, when an ellipse or circle is replaced by a series of line segments. On a high resolution display, the resulting image will appear a lot rougher than it would have appeared had the basic geometric elements been stored in the CGM rather than the straight-line approximations.

The testing program should meet the general requirements set out in the proposed FIPS for Conformance Testing (Reference 8). These issues are discussed further in section VI.

2.0 Impact on the Commercial Marketplace

The results of section III, subsection 4 show that all providers of CGMs will have to develop a new release of their software in order to meet the requirements of any CALS Application Profile. Given, then, that a new release will be required, in this section, an estimate of the amount of resources that the typical implementor will have to expend in order to satisfy the CALS AP is given.

2.1 Constraints on Element Usage and on Options

The most important thing to note is that meeting most the requirements of the AP will be very easy for implementors. Restrictions on the encoding, types, and precisions are easy to build in--either as the only option available or as a restriction controlled by some sort of software "switch," selectable at run-time by the operator of the program generating the metafile. Many other AP constraints--avoiding private ESCAPEs, GDPs, and out-of-range enumerated types--can also be met when the "CALS switch" is set.

Some implementors, especially those using Henderson Software's basic libraries, already are prepared for this sort of thing. They have implemented the constraints of the TOP 3.0 Application Profile for CGM in a similar manner. To minimize the impact on these suppliers and because of the wide publicity concerning the TOP CGM AP, **it may be desirable to have a level of CALS CGM AP conformance be identical to that of TOP.** This is being strongly considered as a change to MIL-D-28003 for FY '89.

Many of the CALS AP constraints and requirements can be implemented with literally a few lines of code. Others are straightforward, but require more effort. For example, if the application generating the CGM uses private (i.e., non-standardized) line types, marker types, or hatch patterns, the CGM generator logic either will have to map the value into a supported CALS value (8 special line types and 18 special hatch patterns are allowed) or will have to simulate the desired visual effect by using other legal CGM elements, like POLYLINE and POLYGON.

In a similar fashion, programs that use GDPs and ESCAPEs to achieve certain visual effects will have to draw upon the supported CGM elements (19 graphical primitives and 35 attributes) to achieve the desired effect. Depending upon the sophistication of the desired effect, the programming effort might be simple or very complex.

2.2 The Color Table and Fonts

In two areas of MIL-D-28003, education of the suppliers is needed: use of the color table and use of fonts. The cooperation that arises when planning for industry demonstrations like CALS EXPO and NCGA's Integrate'89 is a good focal point and opportunity for teaching the implementors what was intended by the Application Profile.

In most cases, proper use of the COLOUR TABLE element will not be expensive to implement: in fact, most products almost work correctly. It's just that, in some of them, no Color Table is written at all and, in others, some color indices are used but

not set. This opens the door to non-predictable results upon interpretation of the CGM. This might be tolerable for Draft Level quality interchange, but it cannot be accepted for Publication Level quality interchange.

Proper use of the elements relating to fonts will require more education and more programming effort to get working correctly. The CGM scheme is based on long-standing ISO and ANSI standards developed by the Codes and Character Set committees. These standards distinguish between a **character set** (a collection of glyphs that comprise an alphabet) and a **font** (the appearance of the characters united by a unique stylistic theme). For now, it is important to get people to use the following elements properly:

FONT LIST
CHARACTER SET LIST
CHARACTER CODING ANNOUNCER
TEXT FONT INDEX
CHARACTER SET INDEX
ALTERNATE CHARACTER SET INDEX

The bad news is that few, if any, CGM generating and interpreting systems have a text environment that utilizes the concepts of ISO 2022, which provides the mechanism for alternately selecting from among several character sets. The good news is that the default values for most elements are consistent with the common text environments familiar to most implementors. Section VI discusses some more suggestions relating to text and the proper selection of fonts.

2.3 Parameter Size Limits

Most of the size limitations (e.g., restricting point arrays to 1024 points) are reasonable and necessary so as to not put an undo burden on interpreters. However, for certain purposes, the constraints on CELL ARRAY and PATTERN TABLE might be too limiting. If one needs to use CELL ARRAY, constraining its use to one 1024 x 1024 array might be much too limiting. However, there is no empirical evidence to support any particular limitation.

The same point holds true for PATTERN TABLE. But, in this case, all those companies that can generate or interpret the PATTERN TABLE element have chosen to permit more than the space occupied by eight 16 x 16 patterns.

These issues are discussed further in section VI, where it is argued that perhaps a special AP level is required for CGMs that use CELL ARRAY and PATTERN TABLE.

2.4 Physical File Structure

The 80-octet fixed length records proposed by MIL-D-28003 are supported by only a very few of the current CGM suppliers. On PCs and on UNIX workstations, the more usual format is an unstructured, continuous stream of octets.

Converting one's program to output 80-character, fixed-length records is not difficult or expensive, but it might encounter some emotional resistance from developers. Perhaps, developers instead will offer a simple utility that takes in a CGM as continuous stream of octets and produces a CGM with the correct physical format.

3.0 Summary

The net impact on the suppliers--from a time and materials point of view--is not very great. Most current CGM generators could be converted to conforming CALS AP generators with less than two man-months effort by the original developers.

One cannot make a similar estimate for CGM interpreters, because the extent of the work required depends upon how rich and robust the current implementation is. All will have to add support for the new CALS line types and hatch patterns and the three new CALS ESCAPEs. Many will have to add support for the 16 Hershey fonts specified in MIL-D-28003 and most will have to implement the full text model (e.g., designating and selecting character sets according to ISO 2022; support for changing text attributes within a sequence of RESTRICTED TEXT, APPEND TEXT, and TEXT elements) assumed by the CGM. Finally, for Publication Level quality conformance, it is likely that nearly all current implementations will have to be improved to meet the guidelines specified in Annex D of the CGM standard and mandated by the CALS AP for conforming interpreters. The rest of the CALS AP requirements are trivial for interpreters to meet, if the interpreter already supports the full set of elements contained in the ANSI standard.

The principal impediment to improved CALS CGM interpreters is that most current interpreters are built upon other, older technology developed before the CGM was adopted as a standard or known to the developers. In order to meet the full requirements of the CALS AP for publication quality interpreters, these products will need to be redesigned and re-implemented with the CGM and CALS requirements firmly in mind from the beginning. This is as it should be: the long-term CALS needs for quality publications involving text and graphics are too important to be sacrificed in favor of the short-term desires of industry to reuse existing code. The Draft conformance level is affordably attainable by those companies seeking to quickly bring a product to market, utilizing pre-CGM technology.

VI. RECOMMENDATIONS

1.0 For CALS Certification of CGMs

As a result of this task, NIST has developed a number of recommendations for CALS. [Note: They are the statements made in bold print in the sections below.]

MIL-D-28003 contains the concepts of a **conforming basic generator** and a **conforming basic interpreter.** A conforming basic interpreter can conform to one of two levels, **Publication** and **Draft.** Certification of conforming basic generators is examined separately from conforming basic interpreters.

2.0 Conforming Basic Generator

Essentially, a conforming basic generator is defined as one that produces only **conforming basic metafiles** (or can be commanded to function in that mode).

In theory, one must test all possible CGMs producible from a generator-under-test in order to verify that all such CGMs are conforming basic metafiles. This is clearly unrealistic. An alternate theoretical approach would be to prove--in some formal sense--that the generator program was incapable of writing a non-conforming basic metafile. This, too, is beyond the state of the art. In fact, any examination of the generator program itself would be prohibitively expensive to administer and would produce, at best, unreliable and unreproducible results.

How then can generator conformance be tested? The only workable approach is to examine a representative sample of the output of the generator-under-test and verify that each CGM is indeed a basic conforming metafile. Consequently, NIST recommends:

> **A reference interpreter capable of testing and reporting whether a CGM-under-test is a CALS basic conforming metafile must be developed.**
>
> If the CGM-under-test fails the test, the report should indicate in what ways the CGM-under-test is non-conforming.
>
> **A standardized form, to be filled out by implementors for their generator-under-test, and a sampling methodology must be developed in order to specify exactly how many CGMs must be provided for the test sample and what the characteristics of those CGMs must be.**
>
> This is a non-trivial task and will require a great deal of careful thought.

3.0 Conforming Basic Interpreter

The only practical way to validate an interpreter-under-test is to provide it with a large set of basic conforming metafiles and observe its behavior and the visible results of interpretation as it attempts to process each CGM. Consequently,

> **CALS needs to acquire a CGM Test Set along with associated test administrator instructions for the validation of CGM interpreters.**
>
> This technique, known as falsification testing, can only prove an implementation non-conforming. Successful processing of all test cases does not guarantee perfect conformance; however, a well chosen test set can boost confidence in the functioning of the interpreter-under-test to very high levels.

Because of the large number of features--both syntactic and semantic--that need to be tested, NIST suggests that

> **A large number of groups of test sets should be developed.**
>
> Each group should concentrate on a few closely-related aspects of conformance. For example, on the use of Metafile Descriptor Elements or on the correct interpretation of the text primitives and attributes. The initial group developed should be a broad collection of CGMs representative of the actual engineering and technical drawings that are expected to be transferred in a CALS application.

Because of the need to tailor the test cases to check for a wide selection of variations and for certain exceptional conditions, NIST also suggests that

> **A CGM creation tool, probably based on the CGM Clear Text Encoding, should be implemented so that needed CGM test sets can be developed in a timely and cost-effective manner.**
>
> This tool must be capable of creating both conforming and non-conforming CGMs in order to verify the proper functioning and robustness of interpreters-under-test.

4.0 Additional Remarks Regarding Certification

Before leaving the topic of certification, NIST makes the following observations:

- Most CGM implementations have been developed in the US and reflect US market imperatives.

- Any certification based on ISO 8632 or ANSI X3.122 alone, while necessary for CALS, is insufficient to guarantee the interoperability required by CALS.

- The CGM has the potential to be a very cost-effective and well-supported standard for the US Government, if an effective certification scheme can be implemented in a timely fashion.

Therefore, NIST concludes:

> **It is vital that the US take the international initiative in the certification and testing of CGM implementations and that the CALS needs for testing against its Application Profile drive the US certification efforts.**

Furthermore, NIST notes that the number and variety of implementations purporting to generate CALS conforming basic metafiles should exceed the number of CALS conforming interpreters needed by the Government and industry and that a greater part of the testing of generators can be automated than the testing of interpreters. Consequently, NIST suggests:

> **Testing for CALS basic conforming metafiles should have higher priority than testing for CALS basic conforming interpreters.**

Essentially, in the short term, it is more important to have a large collection of conforming generators producing a vast selection of basic conforming metafiles for interpreters to attempt to handle than it is to have fully conforming interpreters without a lot of conforming CGMs for them to interpret.

VII. REFERENCES

1. "CALS FY-86 Final Progress Report," National Bureau of Standards Report, November 1986.

2. "CALS FY-87 Final Progress Report," National Bureau of Standards Report, in four volumes, March 1988.

3. "CALS Core Requirements (Phase 1.0)," Coordination Draft, 24 March 1987.

4. "Automated Interchange of Technical Information," MIL-STD-1840A, Draft, 24 March 1987.

5. "Digital Representation for Communication of Illustration Data: CGM Application Profile," MIL-D-CGM, 26 April 1988.

6. "Information Processing Systems--Computer Graphics--Metafile for the Storage and Transfer of Picture Description Information," ANSI/X3.122-1986 and ISO 8632-1987.

7. CGM and CGI: Metafile and Interface Standards for Computer Graphics, Arnold and Bono, Springer-Verlag, Heidelberg, March 1988.

8. "Conformance Testing Policy and Procedures," Notices from the Federal Register, Vol. 53, No. 149 (Wednesday, August 3, 1988).

9. "Conformance Testing of Implementations of Graphics Standards," Working Draft, ISO/IEC JTC1/SC24 N185.

GRAPHICS, CGM MIL-SPEC

Final Phase I.1 CGM MIL-SPEC

CALS SOW TASK 3.1.2.2

FINAL REPORT

CALS SOW TASK 3.1.2.2

FINAL PHASE I.1 CGM MILSPEC

TABLE OF CONTENTS

I. PURPOSE 1

II. BACKGROUND 1
1.0 Overview of CGM 1
2.0 Rationale for Application Profiles 1
3.0 The Necessity of Updating the CALS AP 2
4.0 Present Work 2

III. DISCUSSION 3
1.0 Major Activities 3
1.1 Review and Comparison of Final CALS '87 and TOP Specifications 3
1.2 Evaluation of Initial Experience with CALS AP 3
1.3 Collaboration with TOP Experts 4
1.4 Registration Proposals 4
1.5 Tracking and Evaluation of Other Related Specifications 4
1.6 Evaluation of Functional Extensions 5
1.7 Review Comments by CALS Industry Standards Group 5
1.8 Review Comments from DoD 5
2.0 Specification of Changes to Draft MIL-D-CGM . . . 5
2.1 Examples of Line/Hatch Styles 5
2.2 Errors in the ANSI/FIPS Standard 6
2.3 Order of Metafile Descriptor Elements 6
2.4 Partitioning is Legal for non-MDR Elements . 6
2.5 Editorial Consistency 6
2.6 Fix Typographic Errors 7
2.7 Note about Tape Blocking 7
2.8 Fix Alignment in Table I 7
2.9 Change MILCGM 7
2.10 Fix Numbering of Notes for Table 1 7
2.11 Editorial Improvement 7
2.12 Note Explaining VDC 7
2.13 Delete Hatch Style 8
2.14 Add Linetypes 8
2.15 Clarify View Surface Clearing 8
2.16 Clipping Description 8
2.17 Revise Hershey Font Names 8
2.18 Default Viewport 8
2.19 Additional Generator Constraints 9
2.20 String Length 9
2.21 Text Bundle Table 9
2.22 Reference ANSI X3.122-1986 9
2.23 Rename Minimal Conformance Level 9
2.24 Functionality of Draft Level 9

TABLE OF CONTENTS (Continued)

2.25 Restrict TRANSPARENCY in the Basic Set . . . 10
2.26 Clarify the GDP Sections 10
2.27 Fix Typo in Device Viewport 10
2.28 Requirement for Character Set List 10
2.29 Color Table 10
2.29.1 Minor Changes 10
2.29.2 Major Color Changes 11
3.0 Deviations between TOP and CALS 13

IV. SUMMARY AND CONCLUSIONS 15

V. RECOMMENDATIONS 16
1.0 Future Work 16
2.0 Consolidation with TOP 16
3.0 Further Extensions to the CALS AP 16

APPENDIX A: TOP CGM APPLICATION PROFILE 19
APPENDIX B: MIL-D-28003 CGM APPLICATION PROFILE 35

FINAL PHASE I.1 CGM MILSPEC

I. PURPOSE

Update the draft CGM Application Profile (AP) for CALS based on DoD/Industry Coordination Reviews, and coordination with the TOP Application Profile. Produce final Phase I.1 CGM MILSPEC (CALS SOW Task 3.1.2.2).

II. BACKGROUND

1.0 Overview of CGM

The Computer Graphics Metafile (CGM) standard, ANSI X3.122-1986 and ISO 8632/1-4, specifies the syntax and semantics of a standard file format for storing and communicating computer graphics pictures. By intentional choice of scope the CGM specification is limited to the syntax and semantics of a set of CGM "elements" for the device-independent description of computer graphics pictures.

In the two years that it has been an ANSI standard CGM's use and its incorporation into other standard interface and exchange specifications has been increasing. The number of commercially available implementations continues to increase. For the first time in 1988, test sets for interpreters and output analysis programs for generators have become commercially available. This alleviates one of the problems that has been retarding expansion of CGM support.

CGM has been designated as a Federal Information Processing Standard (FIPS PUB 128). It has been incorporated as the graphical metafile of the TOP (Technical Office Protocol) specification (see Appendix A for the final text of this specification). It is designated as the Geometric Graphic Content Architecture of the ISO (International Standards Organization) compound document standard (ISO 8613, aka "ODA/ODIF"). CGM may be included in SGML (Standard Generalized Markup Language) documents for definition of graphics content. The 1987 CGM Application Profile for CALS was designated as the MIL-D-CGM Military Specification. As of November 1988 it is now known as MIL-D-28003 (see Appendix B for the final text of this specification as of November 18, 1988).

2.0 Rationale for Application Profiles

The syntactic specification of the CGM standard is complete and unambiguous. The semantic specification is less complete. The expected overall results of using the geometric primitive

elements are well enough specified. However some of the finer details, such as the precise appearance of joints and endpoints in lines, are unspecified. This underspecification of semantics was to some extent intended by the bodies of experts who devised the CGM, since it allows a wider range of existing systems to be accommodated and makes the standard more adaptable to the varying needs and philosophies of CGM's diverse clientele.

The semantic ambiguity does mean that there will be no single correct interpretation of a given CGM, and hence it will be difficult to unambiguously describe an intended picture using CGM. The CGM standard also specifically excludes standardization of the behavior of metafile generators and metafile interpreters, thereby introducing a further unpredictability of results into the graphics and application system viewed as a whole. These uncertainties are a distinct drawback in the CGM application areas of Technical Illustration and Technical Publishing, which are central to the CALS effort.

It is primarily to remove these uncertainties that Application Profiles (APs) are defined. One further use of APs is to extend the functionality of the standard where it is deemed inadequate. At the start of the CALS AP project in 1987, one such AP definition was in progress-- the CGM Application Profile of TOP. The NIST AP project progressed in close collaboration with the metafile experts of TOP, with the intention that the two profiles should be identical if possible. An initial specification of a CGM Application Profile for CALS was produced, and that specification was very close to the TOP specification (which was itself extensively revised as a result of the collaborative efforts) in those areas where the two APs overlap. The CALS AP goes further in its specifications.

3.0 The Necessity of Updating the CALS AP

It was recognized at the end of the project for the initial version of the CALS AP that future work would be required and an updated Application Profile should be produced. In part this is necessary because there were not sufficient time and resources to resolve all of the technical issues and the complex set of relationships between the CALS AP effort and other official and de facto standards work; in part it is necessary because events in 1988 provided real world experience and feedback on CGM and the APs; and finally because closely related specifications and standards, with which the CALS AP must be compatible or upon which it depends, have continued to change and evolve in 1988.

4.0 Present Work

During 1988 both of the APs have been reviewed, and there have been real-world demonstrations (e.g. Integrate '88 at the National Computer Graphics Association (NCGA) Conference) of multi-vendor systems integration based on Application Profiles of CGM. Information that has been generated from these and a number of additional sources has pointed to the necessity to revise and extend the initial draft of the CALS AP.

The final report of this project in 1987 contains considerable technical detail about the requirements, scope, and goals of the CALS application profile. It contains as well information on the sources of technical specification for the CALS AP, details of the TOP profile, and the history and process of deriving the CALS AP and reconciling it with the TOP AP. The remainder of this current report will focus on the activities to revise the initial CALS AP in 1988, will present the final results, discuss reconciliation with TOP, and finally identify what needs to be done in 1989 with regard to the CALS AP.

III. DISCUSSION

1.0 Major Activities

The following activities comprise most of the work to date in 1988 for the CALS AP Update project.

1.1 Review and Comparison of Final CALS '87 and TOP Specifications

The final TOP specification became available in April 1988 (see Attachment A). The MIL-D-CGM, which was prepared from the final text of the CALS AP '87, became available in April 1988. All of these documents have been reviewed for correctness and consistency. In addition, review comments from independent experts and other reviewers have been received and considered. Changes to both the TOP AP and MIL-D-CGM resulted.

1.2 Evaluation of Initial Experience with CALS AP

At NCGA '88 there was for the first time a substantial multi-vendor demonstration of system integration using CGM. The TOP and CALS APs were not specifically required of participants. Never-the-less, some of the requirements were based on the APs, some were based on variations of specifications from the APs, and some vendors used TOP/CALS conforming generators and interpreters. In addition, experience has been gained during the real work of building implementations and constructing test sets. All of this experience has been evaluated for lessons about the

strengths and deficiencies of the APs. Significant changes to MIL-D-CGM, particularly in the area of color, have resulted.

1.3 Collaboration with TOP Experts

Consultation and discussion with the experts responsible for the TOP profile has continued and will continue. The TOP profile is soon to be printed in final form. A number of "editorial" changes have been produced (for both TOP and CALS), and a number of substantial changes are being discussed for the TOP revision cycle. Discontinuing independent development of the TOP AP, in favor of a single AP based on the CALS AP '87 and its updates, has been discussed.

1.4 Registration Proposals

Two sets of registration proposals of interest to CALS are being processed. First, there is a set of linetypes and hatch styles from engineering practice. These were included in the CALS AP '87. These have been submitted for Graphical Registration within ISO (with ANSI being the reviewer and submitter). The progress of these proposals has been tracked. Second, a set of registration proposals for extended geometric functionality (conics, splines, etc) that were not included in CALS AP '87 have been submitted for registration. These have been followed and evaluated as well. Changes to MIL-D-CGM have resulted.

1.5 Tracking and Evaluation of Other Related Specifications

CALS AP '87 attempted to specify font naming by reference to ISO 9541. At the time the TOP AP was written (upon which the CALS AP '87 was based), ISO 9541 was at Draft Proposal (DP) stage. Changes have since occurred and are still occurring. These have been studied and evaluated. Changes to both TOP and MIL-D-CGM have resulted--mainly it is considered premature to try to formulate font name specifications according to the unstable DIS (Draft International Standard) ISO 9541.

There is significant overlap between CALS AP '87 and the specification of CGM in ISO 8613 Part 8, which is ODA/ODIF Geometric Graphics Content Architecture. This specification has been tracked and commented on. Minor changes to TOP, CALS, and ODA/ODIF resulted. However ODA/ODIF introduced further changes without consultation or review at the stage of producing final text, and has introduced flaws. These will not be repeated in the CALS and TOP APs.

Finally, the existence of an "implementors agreement" related to 8613/8 (an NBS Document Application Profile) was belatedly

discovered. It is believed to be essentially the same as 8613/8. The previous comments on ODA/ODIF apply here as well.

1.6 Evaluation of Functional Extensions

The second set of registration proposals, mentioned above, and the preliminary content of the CGM addenda (Addendum 1 and Addendum 3) deal with this set of functional extensions, which are important to the CALS constituency. The work is being followed carefully. Strategies for relating and coordinating the AP to the content of this work are being considered. No changes to MIL-D-CGM have resulted so far.

1.7 Review Comments by CALS Industry Standards Group

A CALS industry standards working group met to consider comments on MIL-D-CGM. Comments were received, classified, discussed, and responded to. The dispositions included rejection, acceptance, acceptance with modification, and deferral. The last category were deferred to NIST for processing. All of the comments and dispositions were reviewed as well. In a few cases NIST recommended modifications of the initial dispositions. Changes to MIL-D-CGM resulted from this review.

1.8 Review Comments from DoD

In September 1988 comments on MIL-D-CGM were received from Department of Defense for processing. There was no initial screening of these comments. 30+ comments were processed and dispositions recommended for them. Changes to MIL-D-CGM resulted from processing these comments.

2.0 Specification of Changes to Draft MIL-D-CGM

As a result of the previous activities the following changes are recommended for CAL AP '87--MIL-D-CGM.

2.1 Examples of Line/Hatch Styles

An appendix will be added that gives examples of the engineering line and hatch styles. The hatch style examples are taken from Y14.M. The line type examples are taken from a recent draft of the registration proposals. They are included in Attachment A to this report. When the proposals are finally accepted, the examples should be replaced.

2.2 Errors in the ANSI/FIPS Standard

A number of harmless editorial errors have been found. In addition, a number of error of substance (but due to editorial causes) have been found. These should be included in the updated MIL-D-CGM, as a subsection of Section 6, or as a new Section 7, or as another appendix.

1. ANSI CGM has an error that was corrected in ISO CGM. On p.100, the last item on the page: "1" should be "0" and "foreground" should be "background".

2. In ANSI CGM (and ISO), Part 3, p.17, item 11: the fraction numerator which is "pnx" should be "pnx-1".

3. In ANSI CGM (and ISO), Part 3, p.26, VDC REAL PRECISION: "3I" should be "E,2I".

4. In ANSI CGM (and ISO), Part 1, clause 5.2.1, clause 5.3.12, clause 6: it is unclear and contradictory whether Metafile Descriptor elements return to default at Begin Picture, and whether they can be included in the Metafile Defaults Replacement. The answer is "no" on both points.

5. In ANSI X3.122-1986, Part 1, p.106, the expansion of "<metafile contents>": the "|" symbols should be deleted.

2.3 Order of Metafile Descriptor Elements

It is unclear in the CGM standard whether there should be a mandatory ordering of Metafile Descriptor elements (the grammar implies some). The CGM extension experts feel that the order should be: Metafile Version, Metafile Element List, Metafile Description. CGM Addendum 1 will impose such an ordering. This specification will be put into the AP in a future revision, when the AP is based on Addendum 1. It should be suggested in 3.2.1.2, pointing out that it may be mandatory in a future revision.

2.4 Partitioning is Legal for non-MDR Elements

Add a sentence to the 3.2.7.1 at the end of Description for Metafile Defaults Replacement: "Partitioning is permitted for all other elements."

2.5 Editorial Consistency

Insure that all references to "conforming..." read "conforming basic metafile", or "conforming basic generator", or "conforming basic interpreter", as the case may be. That is, the word order should be "conforming basic <xxx>". See 3.2.8.1 for an example that needs fixing.

2.6 Fix Typographic Errors

In 2.1.1, "availcable" should be available; "fromshall" should be "from"; "department" should be capitalized.

2.7 Note about Tape Blocking

The final text for the TOP AP and the submitted text for the CALS AP '87 both had a second sentence in 3.1.5: "When files are being transmitted on magnetic tape, the 80-octet logical records should be blocked into 800-octet physical records." If this was not removed for a reason, it should be restored. Also, change "application profiles conforming" to "conforming basic metafiles".

2.8 Fix Alignment in Table I

The word "Announcer" should be indented to emphasize that it is part of the previous name, continued on the second line. Or else put it all on one line if it will fit.

2.9 Change MILCGM

The name MILCGM in Note 1 of Table I should be changed to MIL-D-28003.

2.10 Fix Numbering of Notes for Table 1

Move the content of Note 4 to Note 2. Move the content of Note 2 to Note 3. Move the content of Note 3 to Note 4.

2.11 Editorial Improvement

In 3.2.1.3, delete the sentence beginning "This is not an error...".

2.12 Note Explaining VDC

Below table II add "Note: VDC stands for Virtual Device Coordinates, the coordinate system of CGM FIPS 128.

2.13 Delete Hatch Style

Remove "earth" the earth style from Table V, because it has been withdrawn from registration (it can be made with "rock" and "sand"). Note: "across grain wood", "with grain wood", and "water and other liquids" have been reformulated to meet critical objections and have been resubmitted to registration. They are lagging one step behind the others now.

2.14 Add Linetypes

Add to Table IV the two linetypes

break line, style 1 -11309
break line, style 2 -11310

2.15 Clarify View Surface Clearing

Rewrite 3.2.4.1: "Unless clearing is suppressed by 'Escape -301', the view surface shall be cleared upon interpretation of the Begin Picture Body element.

2.16 Clipping Description

Rewrite 3.2.4.1: "When the clip indicator is 'off', clipping shall be done to the intersection of the device viewport and the device view surface limits. When clipping is 'on', clipping shall be done to the intersection of the clip rectangle, the VDC extent, the device viewport and the device view surface limits."

2.17 Revise Hershey Font Names

Throughout, replace "Hersey" with "Hershey". In 3.2.5, replace the three sentences beginning with "[NOTE: The font..." by "Font names will be specified in a manner compatible with ISO 9541, Font and Character Information Interchange, when it becomes stable. At this point the font name is the concatenation of the string "HERSHEY:", to designate one of the Hershey fonts, and a "name string" to designate the particular typeface."

2.18 Default Viewport

Rewrite section 3.2.6.2: "The default device viewport is the largest rectangular area of the screen. The viewport is redefined by this escape element by specifying two corner points. The effective viewport is the largest rectangle in the viewport having the same aspect ratio as the VDC extent, and having the same lower-left corner as the viewport. The VDC extent is mapped onto the effective viewport when the picture is interpreted. The units with which the two points are specified are real fractions [0.0 to 1.0], which shall be applied to the default viewport. If used, this Escape must appear in the Picture Descriptor. If the scaling mode has been set to metric, then the device viewport shall have precedence -- the scaling mode will be treated as if it were abstract.

2.19 Additional Generator Constraints

In 3.2.8.1, change "by MODULO onto the Basic range" to "the default value for that attribute;"

2.20 String Length

In 3.2.8.3, Maximum String Length, change 256 to 254.

2.21 Text Bundle Table

In table VIII, Text Bundle, change character expansion in the second text bundle from 0.5 to 0.7.

2.22 Reference ANSI X3.122-1986

In 2.1.1, mention that FIPS 128 is identical to ANSI X3.122-1986.

2.23 Rename Minimal Conformance Level

In 3.1.2, rename "Minimal Level" to "Draft Level". Make same change in 1.2. Add to 3.1.2, or somewhere, a sentence explaining that Publication Level is intended to be mandatory for final document production, and Draft Level defines a level that is suitable for working when documents are in development or draft stage (Draft Level could be much less expensive and time consuming to produce).

2.24 Functionality of Draft Level

Delete "MARKER TYPE", "LINE TYPE", "HATCH INDEX", "FONT DESIGN" and "EDGE TYPE" from the table at the end of 3.1.2. Add a

sentence after the table: "For LINE TYPE and HATCH INDEX, the interpreter shall have the capabilities of D.5 of FIPS PUB 128. However the interpreter may take fallback actions for the additional linetypes of Table IV and hatch styles of Table V.

2.25 Restrict TRANSPARENCY in the Basic Set

Add to table II a line for the Transparency element, specifying the Basic values to be "1 (on)".

2.26 Clarify the GDP Sections

Rewrite 3.2.1.5: "To ensure portability and predictability of results, conforming basic metafiles shall not contain any Generalized Drawing Primitive (GDP) elements. Future addenda to this profile may specify GDP elements to be included in the Basic set."

2.27 Fix Typo in Device Viewport

In MIL-D-CGM, section 3.2.6.2, the example: "-301" should be "-302".

2.28 Requirement for Character Set List

In 3.2.3, delete the sentence "Each basic conforming...(1, 4/1)".

2.29 Color Table

2.29.1 Minor Changes

In 3.2.3, delete the last two sentences of the paragraph, beginning with "FIPS PUB 128...".

In 3.2.7.1, Color Table, replace the second sentence with "Any Color Table element defining the representation of a given color index shall appear in the picture before reference to that index by an attribute element or use of that index by a graphical primitive element (included in the latter is implicit use of default color index attribute values by the first occurrence of an associated primitive). Once a given color representation is defined and used, it shall not be redefined. These restrictions insure that interpreting systems without dynamic color update capabilities can render the intended picture accurately."

2.29.2 Major Color Changes

It has become clear that the default color specification of the TOP profile and the first draft of the MIL-D-CGM profile not only fails to contribute to predictable results, but in fact positively leads to unpredictable results in some circumstances. Other color problems have been pointed out in the comments on MIL-D-CGM. The color handling under MIL-D-CGM will be revised as follows in an integrated solution to all of these problems.

In section 3.2.1.6, add a paragraph: "For indexed color selection, either all color indexes used in the metafile shall have their representations defined by use of the COLOR TABLE element, or none shall. A color index is "used" if it occurs in an element selecting a color value to be applied to a primitive (LINE COLOR, CELL ARRAY, etc).

Replace first sentence of 3.2.8.2 with: "In the absence of any color table elements, or of ESCAPE -303 (see 3.2.6.3), in the metafile, conforming basic interpreters shall initialize their color tables as follows: index 0 shall be set to white, index 1 shall be set to black, and indexes 2-254 shall be set by cyclic repetition of the 8 entries specified in table VII. Draft conformance level for interpreters allows black and white to be reversed in the first two indices.

In table VII, replace 255 with 1.0, and add a note after the table: "NOTE: The values '1.0' in the preceding table denote full intensity for the appropriate component."

Delete the last sentence of 3.2.8.2.

Add a new 3.2.6.3 Implicit Color Table:

> The default Color Table is undefined in FIPS PUB 128. It is always possible to specify an entire default Color Table with Metafile Defaults Replacement. This ESCAPE element provides a shorthand method to select a default color table from a small set of predefined tables, and provides as well a method to specify that the interpreter shall define its own color table according to available resources. This ESCAPE is allowed in the Metafile Descriptor. If used, no COLOR TABLE elements may appear in the metafile. The single integer parameter of the ESCAPE has the following meanings:
>
> 0: "none" -- there is no implicit default value of the color table, interpreters may associate representations with color indexes as they see fit.
>
> 1: "cyclic" -- the interpreter should initialize its color table to contain
>
> 0 - white

1 - black
2 - red
3 - green
4 - blue
5 - yellow
6 - magenta
7 - cyan
8 - black
9 - white
10 through 255 - cyclic repetition of 2-9.

2: "uniform" -- the interpreter should initialize its color table to: black and white in indexes 0 and 1. Beginning at index 2, 224 representations that uniformly sample the RGB color cube as follows:

- 5 levels of red evenly spaced from none to maximum;
- 9 levels of green evenly spaced from none to maximum;
- 5 levels of blue evenly spaced from none to maximum.

The color cube is mapped to the linear array by varying the red dimension most rapidly, then the green dimension, then the blue dimension. Note that the 225th element implied by this sequence, black, is not put into the table. Beginning at index 226 are 28 levels of gray. Index 226 is 1/32 (very dark). Succeeding entries, with the exceptions described below, are 1/32 lighter. The exceptions are 0/32, 8/32, 16/32, 24/32, 32/32, which may be found in the "color cube" section of the table.

The default value of the parameter is 1, "cyclic".

ESCAPE IDENTIFIER: -303

ESCAPE DATA RECORD: A single string of text containing the single integer parameter of this escape element. The integer is encoded as "clear text", i.e., value 2 is encoded as the string comprised of (or containing) the ASCII character "2".

Rewrite the last section of 3.2.8.1: "Conforming generators shall provide to applications the means to either select the implicit color table for a metafile or to ascertain the value that will be used in the metafile. The means to ascertain the value may consist of either a software inquiry mechanism or documentation accompanying the system. An inquiry mechanism is the preferred method."

3.0 Deviations between TOP and CALS

As specified in the order for this project, liaison with the experts developing and maintaining the TOP CGM Application Profile has been maintained. Changes of interest to both profiles continued to be negotiated until input to the TOP AP was closed and the final version printed (see Appendix A). Some significant issues were raised since the closure of TOP to technical change, during review of MIL-D-CGM within the CALS industry community and within DoD.

Prior to closure of input to the TOP AP, a number of differences exist between the two profiles (and are recognized in the TOP AP):

1. The Metafile Description element in the TOP AP contains the substring TOP/BASIC-1, whereas in MIL-D-CGM it contains the substring MIL-D-CGM/BASIC-1.

2. The TOP AP allows interpreters to use only the Hershey fonts for rendering pictures. MIL-D-CGM allows the use of any font which is "metrically identical" to a requested Hershey font.

3. MIL-D-CGM has added a number of additional linetypes and hatch styles for engineering applications.

4. MIL-D-CGM has specified two conformance levels for interpreters (Publication Level and Draft Level), whereas the TOP AP specified only one.

Since closure, attempts to reconcile review comments on MIL-D-CGM have lead to a number of additional differences of substance:

a. Since CGM Addendum 1 will likely impose an order on a few of the Metafile Descriptor elements, a note to this effect is added to MIL-D-CGM and a recommendation that this order be observed (for the elements Metafile Version, Metafile Element List, and Metafile Description).

b. In MIL-D-CGM the behavior of the Device Viewport escape has been clarified and brought into alignment with the specifications of ISO CGI (Computer Graphics Interface) and the ISO CGM Addendum 1. It is not clear whether this is slightly different from that intended in TOP or simply a more precise statement of the requirement.

c. The maximum string length has been changed in MIL-D-CGM from 256 to 254. 256 was somewhat of an arbitrary choice, and is just slightly longer than the longest

string expressible in a short-format binary text string (254).

d. The Basic values of Transparency in MIL-D-CGM have been limited to the single value 1 (on). The effects of this element in CGM itself are described as device dependent for the value 0 (off).

e. MIL-D-CGM has dropped the requirement that the Metafile Descriptor contain the Character Set List element listing (0,4/2) and (0,4/1). The vast majority of applications will simply use ASCII, and that is the default, so announcing the requirement for the other set makes no sense.

f. Significant revisions have been made to the color functionality in MIL-D-CGM. The differences with TOP now are:

- TOP says all color table elements must appear before the first graphical primitives. MIL-D-CGM now says that they just must appear before they are used by a primitive or attribute. The effect is the same but the latter specification will be somewhat easier for applications and generators.

- There is no requirement for color index definition in the metafile in the TOP AP. MIL-D-CGM now requires that either all indexes which are used in a metafile shall be defined, or none shall be defined. The case of having some indexes which are used be defined and some not defined is prohibited.

- The TOP AP defines the default Color Table for interpreters to be a cyclic repetition of the colors Red, Green, Blue, Yellow, Magenta, Cyan, Black, White, starting at index 2. Indexes 0 and 1 are left undefined. In the default case MIL-D-CGM now specifies the same table except that index 0 is defined to be white and index 1 black.

- MIL-D-CGM has added an Escape element, Implicit Color Table, with Escape Indicator -303. This element allow a generator to specify how interpreters are to initialize their color tables. One option is "none", which means the interpreter is to use its native color capabilities as best it can and initialize its color table accordingly. The other two options are shorthands for selecting one of two useful initialization schemes \(em the first is the "cyclic" scheme (the default for the

AP), and the second is a uniform sampling of the RGB color cube.

IV. SUMMARY AND CONCLUSIONS

In 1987 this project produced the first draft of the CALS CGM Application Profile. This work was carried out in close collaboration with the TOP graphics experts, and in consequence significant changes to the TOP AP were made. The two APs were nearly identical in areas of overlap. Neither profile had been subjected to "field testing", i.e., use in real world applications or demonstrations of multi-vendor systems integration.

In early 1988, the MIL-D-CGM was derived from the CALS AP and incorporated into MIL-STD-1840A. The CGM standard and the CALS and TOP APs received significant attention and use in a number of areas. In addition, the initial drafts of both AP were subject to review by a much wider community during 1988 than had been the case in 1987. Additional real world experience was gained in the process of examining and handling hundreds of CGMs during the production of the first test set for CGM interpreters. Significant progress was made in standardization of CGM extensions and in Graphical Registration for CGM extension.

In consequence of these activities a number of changes have been identified for the CALS AP (MIL-D-CGM). Firstly, there are numerous editorial changes and corrections. There are, in addition, several changes which are mainly clarification but do change the functionality at least slightly (clipping, device viewport, and interpreter conformance levels). A section has been added to document those errors which have been found in the CGM standard itself (FIPS PUB 128). Examples of the engineering linetypes and hatch styles have been added.

Next there are a number of minor functional changes: the contents of the text bundle table are adjusted slightly; maximum string length is changed; there is no need for the character set list element; the tables of additional linetypes and hatch styles are modified slightly; the use of the device dependent Transparency element is restricted.

One of the biggest problems in predictable interchange with CGM has been the use of indexed color. Neither the CALS nor the TOP APs solved this. In fact the specifications in the APs turned out to actually introduce unpredictability into CGM interchange. A set of changes has been recommended to fix these problems and address other comments received during MIL-D-CGM review. Basically, a conforming metafile must either define all color indexes that are used, or define none of them; and a new escape

is added to allow a shorthand definition of the implicit (default) color table for the interpreter.

A handful of changes are recommended to MIL STD 1840A. That specification, and not the MIL-D-CGM, should handle presentation of metafiles in a document (e.g., by reference to ISO 8613); and it should handle physical data formats for metafiles. The binary encoding is felt to be adequate for **interchange**, but 1840A should recommend that delivered CGM software systems (generators and interpreters) have Clear Text capability.

V. RECOMMENDATIONS

1.0 Future Work

A first usable version of the CALS AP is now complete and has been reviewed see Appendix B). Over the next year, two important activities for the CALS AP should be pursued.

2.0 Consolidation with TOP

Because of different production schedules, production and review processes, and editors it has been difficult to keep the TOP and CALS profiles in substantial alignment. In particular, in the last part of the 1988 CALS project the AP was reviewed by a wider audience than had formerly done so, and a number of good comments were received. Some of these led to substantial changes. As a result the two profiles have now diverged somewhat.

This coordination problem will continue to be difficult. In 1989 CALS should pursue with TOP the possibility of merging the two profiles into a single one, with a single technical editor, and reviewing that single profile jointly in the two communities. It appears from initial discussions that there are sufficiently similar constituent requirements to make this feasible.

3.0 Further Extensions to the CALS AP

The current CALS AP has begun the process of extending the functionality of CGM. Additional Linetypes and Hatch Styles, as used in engineering, have been added. These proposals were submitted (under another project) to the Graphical Registration process and have undergone at least two rounds of review. It is expected that they will be approved in substantially their current form, and are therefore stable enough to include in the CALS AP.

There are currently several sources of additional functionality of interest to the CALS constituency. The additional

functionality includes such areas as improved text and font capabilities, advanced geometric primitives (splines, conics, etc), and global segments and symbol libraries. It is particularly critical that improved text and font capabilities be incorporated, as there is no adequate substitute or work-around available in CGM or the APs currently.

The sources for such extensions, and the time frame in which they are expected to reach substantial stability, are:

- The functional extensions of ISO CGM Addendum 1 (1 year);
- Further functional extension through the Graphical Registration process (1-1.5 years);
- The functional extensions of ISO CGM Addendum 3 (2-3 years).

Addendum 3 is being specifically formulated for the needs of technical constituencies such as CALS. It was originally anticipated that the functionality therein would be progressed and stable in a much shorter time frame, and that the CALS AP could wait for such stabilization rather than endorsing variations such as would be found in registration. The actual time frame is now too long to wait.

Consequently in 1989 an addendum to the CALS AP should be formulated to incorporate that functionality which is needed and is then available from Addendum 1 and Graphical Registration. During 1989 Addendum 1 should reach sufficient stability in the ISO pipeline that any of its functionality can be used as appropriate. Many of the proposals of Graphical Registration should similarly have stabilized. Although the content of Addendum 3 may differ in some details from the registration proposals, the basic technology covered will be substantially the same. Having the technology available in less than the 3-year time frame of Addendum 3 now appears to outweigh the disadvantage of implementors having to make changes of detail when the content of Addendum 3 finally becomes standard.

This page left intentionally blank.

APPENDIX A

FINAL TEXT OF CGM

APPLICATION PROFILE FOR TOP

6.2 Computer Graphics Metafile

This section defines the TOP CGM Application Profile (AP) for the Computer Graphics Metafile Interchange Format Building Block. The functional description and binding rules for this building block are found in Chapter 2, Section 2.4.10.

6.2.1 CGM Introduction

The TOP CGM AP defines the conformance characteristics or permissible combinations for all possible data streams that are specified in the profile. In addition, the TOP CGM AP defines additional requirements for transmitting, receiving, interpreting and handling valid CGM data streams. The definition of such implementation constraints is usually outside the scope of an ISO standard. However, such APs are required and necessary to insure uniform implementation of such standards, especially where interchange in an open system environment is concerned.

6.2.2 CGM Scope

The TOP CGM AP defines the CGM implementation that is required for computer graphics picture information interchange. CGM implementations that conform to this AP will be able to be integrated into other application processes such as compound document interchange. This AP can, in the future, be supplemented by additional CGM APs.

6.2.3 Definitions

APPLICATION PROFILE - A specification that defines the use of an International Standard, with a definition of all possible data streams that conform to that profile. An AP insures interoperability of implementations of an International Standard.

BASIC VALUE - The subset of permissible values for parameters of a CGM element that are mandatory for conformance to this AP.

CGM MI - A CGM metafile input workstation.

CGM MO - A CGM metafile output workstation.

COMPOUND DOCUMENT - A digital analog of a document containing more than one component objects (such as character, computer graphics, image or facsimile data).

COMPOUND DOCUMENT INTERCHANGE FORMAT - The specification for a mechanism for storing and transferring a compound document. Refer to ISO 8613.

COMPUTER GRAPHICS INTERFACE (CGI) - The specification for interface techniques with graphical devices.

COMPUTER GRAPHICS METAFILE (CGM) - The specification for a mechanism for storing and transferring picture description information. Refer to ISO 8632.

DATA INTERFACE - The communication boundary (i.e., interface) between software modules or devices comprising one or more operation codes and data (as contrasted with a subroutine call interface).

DEFAULT VALUE - The implicit value for a parameter of a CGM element. For example, default Metafile Name in Begin Metafile element is a null string.

DEVICE DRIVER - The device-dependent portion of a graphics system which supports a physical device. The device driver generates device class specific output.

GRAPHICAL KERNAL SYSTEM - A standardized application programmer's interface to graphics systems. Refer to ISO 7942.

METAFILE - Synonymous with CGM. A representation for the storage and transfer of graphical data and control information. This information contains a device-independent description of one or more pictures.

METAFILE GENERATOR - Synonymous with CGM Generator. The software or hardware that creates a picture or conveys information in the CGM representation.

METAFILE INTERPRETER - Synonymous with CGM Interpreter. The software or hardware that reads the CGM and interprets the contents.

PERMISSIBLE VALUES - The range of valid values for a parameter of a CGM element a specified in ISO 8632.

VIRTUAL DEVICE - An idealized computer graphics device that presents a set of graphics capabilities to graphics software of systems via the CGI.

NOTE: Refer to ISO 8632, clause 3 and ISO 7942, clause 3, for further definitions of computer graphics terms.

6.2.4 CGM Architectural Concepts

The CGM is designed to be usable and useful to a wide range of applications, graphics systems, and devices or workstations. The CGM is graphics system independent, as well as device independent. The CGM is created by a CGM Generator. The CGM Generator resides at the level of the device driver and is invoked by the application callable layer. The CGM Generator can be used to record device-independent picture descriptions, conceptually in parallel with the presentation of images on actual devices. Figure 6.2-1 illustrates the TOP Graphics Reference Model for creation of the CGM.

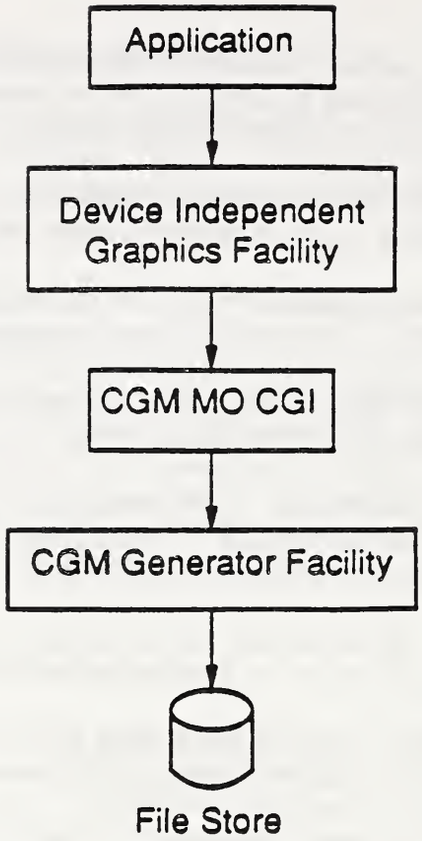


Figure 6.2-1: CGM Generator Reference Model

The CGM is designed to be interpreted in one of two ways. First, the CGM can be interpreted by a special application program, that in turn invokes a device-independent graphics system to render the CGM. Second, a device-independent graphics system may have functions that can be invoked by an application to get, read, and interpret metafile elements using the facilities of a CGM metafile input workstation. Figure 6.2-2 illustrates a CGM Generator (primary) Reference Model. Figure 6.2-3 illustrates the CGM Interpreter (alternate) Reference Model.

The GKS may be the device independent graphics system that is used in figures 6.2-2 and 6.2-3. GKS (ISO 7942), however, does not specifically refer to the CGM any more than it does to another specific class of graphics device.

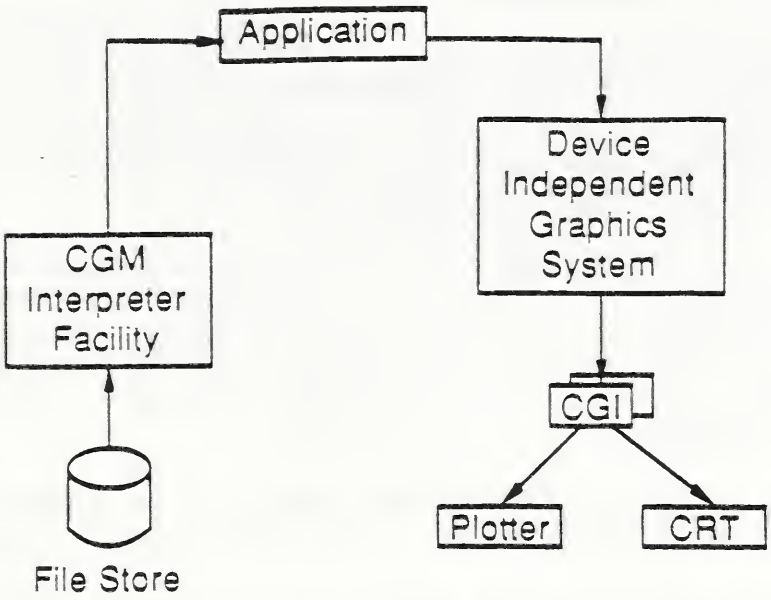


Figure 6.2-2: CGM Interpreter Reference Model

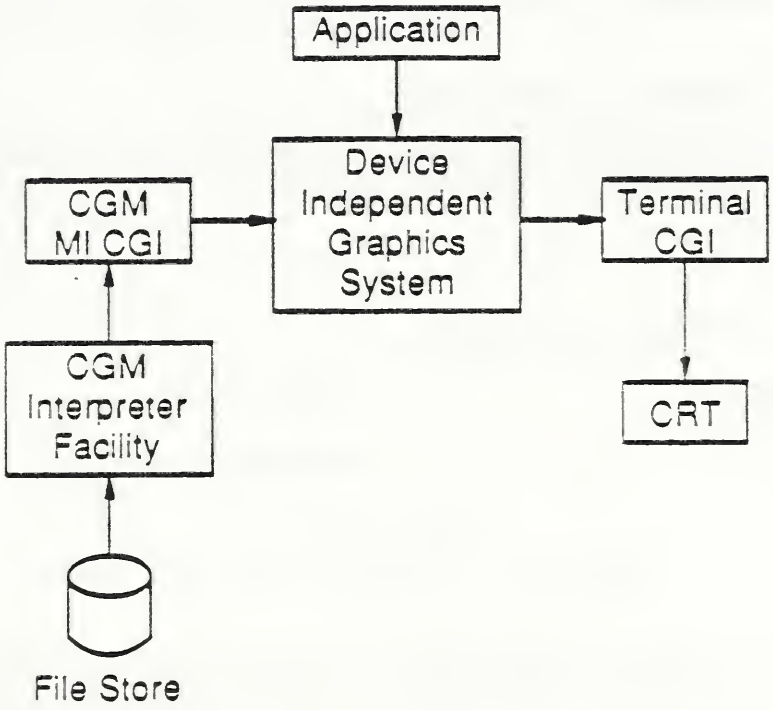


Figure 6.2-3: Alternate CGM Interpreter Reference Model

6.2.5 CGM Conformance

The TOP CGM AP specifies conformance in terms of "permissible and "basic" values. Permissible values are the range of values of CGM elements as specified in ISO 8632. Basic values are a subset of the permissible values that constitute the "basic set". For example, permissible values of LINE TYPE include all non-zero integers, while basic values include the standardized enumerated values 1 to 5.

TOP defines a conforming "basic metafile" to be one that contains no elements or parameter values outside of the basic set. TOP defines a conforming "basic interpreter" to be one that correctly interprets any conforming basic metafile and may have more capability as well. TOP defines a conforming "basic generator" as one that produces only conforming basic metafiles, or can reliably be directed to function in a mode of producing basic metafiles.

In addition, any TOP basic interpreter should correctly parse and pass over any elements that it does not support and any parameter values that it does not support.

CGM (ISO 8632) defines the form (syntax) and the functional behavior (semantics) of the ordered set of metafile elements. There are three different encodings of the CGM that have been standardized. These include Clear Text Encoding, Character Encoding, and Binary Encoding. This AP specifies the CGM Binary Encoding, ISO 8632/3. Future application profiles may be developed for the other encodings.

For interchange of CGM files on a TOP network the binary encoding is required.

6.2.6 Metafile Constraints

The basic set is defined by the limitations on permissible values below. Where an element is not mentioned, it is implied that the basic set includes all values permitted in the CGM.

6.2.6.1 Delimiter Elements

Element	Basic Value
NO-OP	An arbitrary sequence of n octets, n=0,1,2,...,32767

Table 6.2-1: Delimiter Element Constraints

6.2.6.2 Metafile Descriptor Elements

Element	Basic Value
Metafile Description	(Note 1)
Integer Precision	16
Real Precision	0,9,23 (floating point)
	1,16,16 (fixed point)
Index Precision	16
Color Precision	8,16
Color Index Precision	8,16
Maximum Color Index	255
Font List	(Note 4)
Character Set List	0,4/2 (Note 2)
	1,4/1 (Note 3)
Character Coding Announcer	0,1

Note 1: Implementors are encouraged to use the Metafile Description element's string to include a brief identification of their company or product, so that interpreters can account for known idiosyncrasies of generators. The string "TOP/BASIC-1" shall be included within the Metafile Description string to label the metafile as conforming to this profile.

Note 2: The character set is ANS X3.4, 7-bit American National Standard Code for Information Interchange (7-bit ASCII).

Note 3: The character set is ANS X3.134/2, 8-bit American National Standards Standard Code for Information Interchange (8-bit ASCII). This is equivalent to ISO 8859/1, Right-Hand Part of Latin Alphabet Part 1.

Note 4: Four simultaneous fonts are supported. The font names are selected from the basic font names in Table 6.2-8.

Table 6.2-2: Metafile Descriptor Element Constraints

6.2.6.3 Picture Descriptor Elements

Element	Basic Value
Scaling Mode	(Note 1)

Note 1: Implementors should use care in specifying the value of the metric scaling factor to ensure that it has sufficient significant resolution to specify the intended accuracy.

Table 6.2-3: Picture Descriptor Element Constraints

6.2.6.4 Control Elements

Element	Basic Value
VDC Integer Precision	16,32
VDC Real Precision	0,9,23 (floating point) 1,16,16 (fixed point)

Table 6.2-4: Control Element Constraints

6.2.6.5 Graphics Primitive Elements

To ensure portability and predictable results, TOP conforming basic metafiles may not contain any Generalized Drawing Primitive (GDP) elements.

6.2.6.6 Attribute Elements

Element	Basic Value
Line Bundle Index	1 - 5
Line Type	1 - 5
Marker Bundle Index	1 - 5
Marker Type	1 - 5
Text Bundle Index	1 - 2
Text Font Index	1 - 4
Character Set Index	1 - 2
Alternate Character Set Index	1 - 2
Fill Bundle Index	1 - 5
Hatch Index	1 - 6
Pattern Index	1 - 8
Edge Bundle Index	1 - 5
Edge Type	1 - 5
Pattern Table	Pattern Table Index, 1-8 nx, 1-16 ny, 1-16
Color Table	Starting Color Index, 0-255

Table 6.2-5: Attribute Element Constraints

6.2.6.7 Escape Elements

To ensure portability and predictable results, TOP conforming basic metafiles may contain only those ESCAPE elements that are defined in Section 6.2.8.2 of this AP.

6.2.6.8 External Elements

Element	Basic Value
Message	Action Required Flag, 0

Table 6.2-6: External Element Constraints

6.2.7 CGM Defaults

The CGM specifies a complete set of defaults. In a few cases, these defaults are not appropriate for TOP requirements. However, any TOP metafile must be a legal CGM. This includes implicit defaults specified in ISO 8632/1, clause 6, and ISO 8632/3, clause 8. Therefore, each deviation from the implicit defaults requires that the affected element either:

1. Appear in the Metafile Defaults Replacement element, or

2. Be explicitly specified for its value to be applicable.

Each TOP conforming basic metafile shall contain in the Metafile Descriptor a Metafile Defaults Replacement element that includes at a minimum:

1. Text Precision element, where precision is set to 2 (stroke).

Each TOP conforming basic metafile shall also contain in the Metafile Descriptor, the following elements:

1. Maximum Color Index element, where the maximum color index is set to 255.

2. Character Set List element, where the first two character set indices are set to (0,4/2) and (1,4/1).

It is not apparent in the CGM standard what the default value for the precision of the floating point real parameter of Scaling Mode should be. TOP conforming generators and interpreters shall assume that the real precision for this parameter is (0,9,23).

Color table defaults for color indices 0 and 1 are explicitly defined in the CGM standard as corresponding to the nominal background and nominal foreground colors, respectively.

TOP conforming CGM interpreters will initialize their color table with the starting color table index set to 2, and the list of direct color values for the remaining 254 entries a repetition of the following eight values:

Index	Values	Meaning
2	(255, 0, 0)	Red
3	(0, 255, 0)	Green
4	(0, 0, 255)	Blue
5	(255, 255, 0)	Yellow
6	(255, 0, 255)	Magenta
7	(0, 255, 255)	Cyan
8	(0, 0, 0)	Black
9	(255, 255, 255)	White

Table 6.2-7: Default Color Table

6.2.8 CGM Related Private Use of Elements

6.2.8.1 Fonts

The fonts in Table 6.2-8 are public domain fonts, available from the U.S. National Bureau of Standards (NBS) [CGMRef5]. All of these fonts are considered to be basic capabilities of a TOP conforming basic metafile. Any of these fonts may appear in the Font List element in a CGM that conforms to this AP. The font names are specified in a manner compatible with ISO 9541, Font and Character Information Interchange [CGMRef11]. The font name (Font Identifier for Base Font) is a concatenated string of the Universal Font Name and a User Readable Font Name. The Universal Font Name for these fonts is assumed to be "NBS", pending the registration of the National Bureau of Standards with an Organization Name. The User Readable Font Name is the concatenated string "HERSHEY:", to designate one of the Hershey fonts, and "name string", to designate the particular typeface.

Note: The Hersey "fonts" are really combined specifications of font and character set, in the terminology of standards. So support of the Hersey "fonts" really implies support of a number of fonts and character sets.

1. NBS HERSHEY:CARTOGRAPHIC_ROMAN
2. NBS HERSHEY:CARTOGRAPHIC_GREEK
3. NBS HERSHEY:SIMPLEX_ROMAN
4. NBS HERSHEY:SIMPLEX_GREEK
5. NBS HERSHEY:SIMPLEX_SCRIPT
6. NBS HERSHEY:COMPLEX_ROMAN
7. NBS HERSHEY:COMPLEX_GREEK
8. NBS HERSHEY:COMPLEX_SCRIPT
9. NBS HERSHEY:COMPLEX_ITALIC
10. NBS HERSHEY:COMPLEX_CYRILLIC
11. NBS HERSHEY:DUPLEX_ROMAN
12. NBS HERSHEY:TRIPLEX_ROMAN
13. NBS HERSHEY:TRIPLEX_ITALIC
14. NBS HERSHEY:GOTHIC_GERMAN
15. NBS HERSHEY:GOTHIC_ENGLISH
16. NBS HERSHEY:GOTHIC_ITALIAN

There are four parameters. Invalid parameters will result in this Escape element being ignored.

P1: First corner x-coordinate. Real fraction of the default Device Viewport, in the range [0.0, 1.0].

P2: First corner y-coordinate. Real fraction of the default Device Viewport, in the range [0.0, 1.0].

P3: Second corner x-coordinate. Real fraction of the default Device Viewport, in the range [0.0, 1.0].

P4: Second corner y-coordinate. Real fraction of the default Device Viewport, in the range [0.0, 1.0].

For example, a Device Viewport equal to the upper right quarter of the default Device Viewport would be coded with the following Escape element:

Escape Identifier: -3 0 2
Escape Data Record: ".5,.5,1.,1."

OR

Escape Identifier: -3 0 2
Escape Data Record: "0.50 0.50 1.0 1.0"

A Device Viewport equal in width to the left one tenth of the default Device Viewport and equal in height to the default Device Viewport would be coded with the following Escape element:

Escape Identifier: -3 0 2
Escape Data Record: "0., 0., .1, 1."

6.2.9 CGM Implementation Dependencies

This section describes the implementation dependencies and environmental constraints for this AP. Specifying the nominal values for such implementation practices, defaults, and options will facilitate uniform generation and interpretation of the CGM.

6.2.9.1 General Guidelines for CGM Elements

Unless otherwise noted in this AP, all of the guidelines of ISO 8632, Annex D, shall be adhered to by TOP CGM generators and interpreters. In particular, the interpreter minimum capabilities of ISO 8632, Annex D.5, plus the interpreter functions defined in Section 6.2.6, should be the minimum supported capabilities.

Name: Metafile Defaults Replacement

Description: The Metafile Defaults Replacement element shall not be partitioned. In addition, no part of the element will be partitioned.

Note: Multiple occurrences of the Metafile Defaults Replacement element are allowed by the CGM standard and this AP.

Name: Restricted Text

Description: Minimal capability of a basic conforming TOP interpreter shall render the complete restricted text string (i.e., Append Text elements permitted), scaled isotropically (i.e., specified aspect ratio for the text is not distorted), such that the text string fits into the Text Extent parallelogram.

Name: Color Table

Description: The Color Table element has an unspecified effect when it appears in a picture, subsequent to any graphical primitive elements. The Color Table element should appear prior to any graphical primitive elements to insure that interpreting systems without dynamic color update capabilities can render the intended effect.

6.2.9.2 Implementation Guidelines for Generators and Interpreters

This section is meant to augment ISO 8632/1, Annex D.5 and ISO 8632/3, clause 8.

6.2.9.2.1 Minimum Data Structure Support

Name: Maximum Color Array Dimension

Description: The basic value for the number of color values that can appear in a color array or color list parameter. CELL ARRAY has a color list parameter. PATTERN TABLE has a color array parameter. COLOR TABLE has a color list parameter.

Basic Value: 1048576 for CELL ARRAY (i.e., one 1024 x 1024 image),
2048 for PATTERN TABLE (i.e., eight 16 x 16 patterns),
256 for COLOR TABLE (i.e., entries 0-255)

Name: Maximum Point Array Length

Description: The basic value for the number of points that can appear in parameters for metafile elements.

Basic Value: 1024

Name: Maximum String Length

Description: The basic value for the length of an individual string of characters.

Basic Value: 256 for all string parameters except data records,

32767 for data records

Name: Bundle Table

Description: The bundle representations are not settable in the current version of the CGM. This implementation dependency detracts from the open interchange of the CGM. The following default bundle table values will permit a picture to be uniformly rendered by all conforming basic TOP interpreters.

Basic Value: Refer to Table 6.2-9

Bundle Type Bundle Representation	Bundle Index 1	2	3	4	5
Line Bundle					
Line Type	Solid	Dash	Dot	Dash-dot	Dash-dot-dot
Line Width	1	1	1	1	1
Line Color	1	1	1	1	1
Marker Bundle					
Marker Type	Dot	Plus	Asterisk	Circle	Cross
Marker Size	1	1	1	1	1
Marker Color	1	1	1	1	1
Text Bundle					
Font Index	1	1			
Text Precision	Stroke	Stroke			
Character Expansion Factor	1	0.5			
Character Spacing	0	0			
Text Color	1	1			
Fill Area Bundle					
Interior Style	Hatch	Hatch	Hatch	Hatch	Hatch
Fill Color	1	1	1	1	1
Hatch Index	1	2	3	4	5
Pattern Index	1	1	1	1	1
Edge Bundle					
Edge Type	Solid	Dash	Dot	Dash-dot	Dash-dot-dot
Edge Width	1	1	1	1	1
Edge Color	1	1	1	1	1

Table 6.2-9: Basic Bundle Table

6.2.9.2.2 CGM Transfer Format

Operating system dependencies for file formats can often be more of a burden on interoperability than differences in interchange formats. To ensure CGM interoperability, some conventions for file formats are required.

The file containing the CGM should be formatted into fixed length 80 octet records. If the record length is optionally less than 80 octets, even octet records are required.

Note: When the files are being transferred on magnetic tape, 80 octet records should be formatted into blocks of 800 octets.

6.2.10 CGM Error Processing

A TOP conforming interpreter should gracefully recover from any exception condition. If there is something which is not processable by the interpreter, processing of the metafile should continue with the metafile element following that which caused the exception, if possible. Exact details for exception handling are outside the scope of this recommendation.

6.2.11 CGM Conformance Testing

Conformance testing recommendations for the conforming TOP basic metafile will be addressed by subsequent releases of this recommendation.

6.2.12 CGM References

These references relate to documents applicable to this specification.

CGMRef1 ANS X3.4 - 1986, 7-bit American National Standard Code for Information Interchange

CGMRef2 ANS X3.41 - 1974, American National Standard Code Extension Techniques for Use With the 7-bit Coded Character Set of American National Standard Code for Information Interchange

CGMRef3 ANS X3.134/1, American National Standard Code for 8-bit ASCII Structure

CGMRef4 ANS X3.134/2, American National Standard Code for 7-bit and 8-bit ASCII Supplemental Multilingual Graphic Character Set

CGMRef5 NBS Special Publication 424 - April 1976, Hershey Fonts

CGMRef6 ISO 8613, Information Processing - Text and Office Systems - Office Document Architecture and Interchange Format (ODA/ODIF)

CGMRef7 ISO 8632/1, Information Processing Systems - Computer Graphics Metafile (CGM) for the Storage and Transfer of Picture Description Information - Part 1; Functional Specification

CGMRef8 ISO 8632/2, Information Processing Systems - Computer Graphics Metafile (CGM) for the Storage and Transfer of Picture Description Information - Part 2; Character Encoding

CGMRef9 ISO 8632/3, Information Processing Systems - Computer Graphics Metafile (CGM) for the Storage and Transfer of Picture Description Information - Part 3; Binary Encoding

CGMRef10 ISO 8632/4, Information Processing Systems - Computer Graphics Metafile (CGM) for the Storage and Transfer of Picture Description Information - Part 4; Clear Text Encoding

CGMRef11 ISO 9541, Information Processing Systems - Font and Character Information Interchange

CGMRef12 ISO DIS 8859/1, Information Processing - 8-bit Single Byte Coded Graphical Character Sets - Part 1; Latin Alphabet Part 1

CGMRef13 ISO 7942, Information Processing Systems - Computer Graphics - Graphical Kernal System (GKS) Functional Description

CGMRef14 ISO DIS 8805, Information Processing Systems - Computer Graphics - Graphical Kernal System for Three Dimensions (GKS-3D) Functional Description

CGMRef15 ISO DP 9592, Information Processing Systems - Computer Graphics - Programmers Hierarchical Interactive Graphics System (PHIGS)

CGMRef16 ISO TC97/SC21 N1179, Information Processing Systems - Computer Graphics - Interfacing Techniques for Dialogues with Graphical Devices (CGI)

6.2.13 The Use of OSI Data Transfer Services

To transfer a CGM file between two TOP End Systems, the services provided either by FTAM or by MHS can be used. Remote access to part of a CGM file is not addressed at this time.

Using FTAM to transfer CGM files:

> One should specify the CGM file Document Type entry number as FTAM-3, Document-Type-Name as '{ISO standard 8571 document-type (5) unstructured-binary (3)}' for Contents-Type-Attribute or Contents-Type-List. The contents of the CGM file should be mapped onto a sequence of octet strings. The boundary of the octet strings has no significant meaning.

Note: As there is no standard defined document type at this time for a CGM type file, the presentation layer facilities can not be fully used and it is left up to the user or application programs that remotely access files using FTAM to know that a given file contains CGM formatted information.

Using MHS to transfer CGM files:

One should specify the Body as USABodyParts BodyPartNumber '1', and the contents of the CGM file is mapped in the body of an IMP as a sequence of octet strings. The boundary of the octet strings has no particular semantics.

APPENDIX B

FINAL TEXT OF MIL-D-28003
MILITARY SPECIFICATION

DIGITAL REPRESENTATION FOR COMMUNICATION OF ILLUSTRATION DATA:

CGM APPLICATION PROFILE

(As of November 18, 1988)

This page left intentionally blank.

NON-MEASUREMENT SENSITIVE

MIL-D-28003

MILITARY SPECIFICATION

DIGITAL REPRESENTATION FOR COMMUNICATION OF ILLUSTRATION DATA: CGM APPLICATION PROFILE

NOTE: This draft, dated 18 November 1988, prepared by the OSD CALS Office has not been approved and is subject to modification.

DO NOT USE PRIOR TO APPROVAL. (Project ILSS-0034)

1. SCOPE

1.1 Scope. This military specification establishes the requirements to be met when 2-dimensional picture description or illustration data that is predominantly vector is delivered in the digital format of the Computer Graphics Metafile (CGM) as specified by its Federal Information Processing Standard, FIPS PUB 128.

Beneficial comments (recommendations, additions, deletions) and any pertinent data which may be used in improving this document shall be addressed to: Director, CALS Policy Office, DASD(S)CALS Pentagon, Room 2B322, Washington, DC 20301, by using the self addressed Standardization Document approval Proposal (DD Form 1426) appearing at the end of this document or by letter.

AMSC N/A AREA ILSS

DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

1.2 Classification. This specification establishes the requirements for the communication or interchange of illustration data in digital format for use in technical illustrations and publications. The CGM Application Profile (AP) defined by this specification consists of three parts: the metafile, the generator, and the interpreter. There shall be only one level for the metafile and generator, and they shall be called conforming basic metafile and conforming basic generator, respectively. The interpreter shall be one of the following two levels as specified by the contract or purchase order:

Level 1 - Publication Level

Level 2 - Draft Level

Publication Level shall be mandatory for final document production. Draft Level defines a level that is suitable for working when documents are in development or draft stage. Draft Level shall be used for all documents except those for final document production. Additional classes of conforming basic metafiles are expected to be added in future versions of this specification as soon as technical work codifies their requirements and validates fitness for use.

2. APPLICABLE DOCUMENTS

2.1 Government documents.

2.1.1 Specifications and standards. The following standards form a part of this document to the extent specified herein. Unless otherwise specified, the issues of these documents are those listed in the issue of the Department of Defense Index of Specifications and Standards (DODISS) and supplements thereto, cited in the solicitation.

STANDARDS

FEDERAL

FIPS PUB 128 - Computer Graphics Metafile (CGM)

Note: FIPS PUB 128 adopts ANSI X3.122 as a Federal Information Processing Standard.

(Copies of the referenced Federal Information Processing Standards (FIPS) are available to Department of Defense activities from the Commanding Officer, Naval Publications and Forms Center, 5801 Tabor Avenue, Philadelphia, PA 19120-5099. Others must request copies of FIPS from the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.)

MILITARY

MIL-STD-1840 - Automated Interchange of Technical Information

(Copies of the referenced military standard are available from the Department of Defense Single Stock Point, Commanding Officer, Naval Publications and Forms Center, 5801 Tabor Avenue, Philadelphia, PA 19120.)

2.1.2 Other Government documents. The following other Government document forms a part of this document to the extent specified herein. Unless otherwise specified, the issue is that cited in the solicitation.

NATIONAL BUREAU OF STANDARDS

NBS SP 424 - Contributions to Computer Typesetting Techniques: Tables of Coordinates for

Hershey's Repertoire of Oxidental Type Fonts and Graphic Symbols, NBS Special Publication 424, April 1976.

(Application for copies shall be addressed to the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.)

2.2 Non-Government publications. The following documents form a part of this document to the extent specified herein. Unless otherwise specified, the issues of the documents which are DoD adopted are those listed in the issue of the DODISS cited in the solicitation. Unless otherwise specified, the issues of documents not listed in the DODISS are the issues of the documents cited in the solicitation (see 6.2).

NATIONAL STANDARDS

ANSI X3.4 - 7-bit American National Standard Code for Information Interchange (7-bit ASCII)

ANSI X3.134/2 - 8-bit American National Standards Code for Information Interchange (8-bit ASCII)

(Application for copies shall be addressed to: American National Standards Institute, Inc., 1430 Broadway, New York, NY 10018).

(Nongovernment standards and other publications are normally available from the organizations which prepare or which distribute the documents. These documents also may be available in or through libraries or other informational services.)

2.3 Order of precedence. In the event of a conflict between the text of this specification and the references cited herein, the text of this specification shall take precedence. Nothing in this specification, however, shall supersede applicable laws and regulations unless a specific exemption has been obtained.

3. REQUIREMENTS

3.1 General requirements. This specification defines conformance of a CGM metafile in terms of "permissible" and "basic" values. Permissible values are the range of values of CGM elements as specified in FIPS PUB 128. Basic values are a subset of the permissible values and they constitute the "Basic Set." For example, permissible values of MARKER TYPE include all non-zero integers, while basic values are limited to the specific values 1 to 5. A conforming basic metafile shall contain no elements or parameters outside of the Basic Set. The CGM AP which corresponds to the illustration data to be communicated shall be in the form of one or more conforming basic metafiles.

3.1.1 Conforming basic generator. A conforming basic generator shall be defined to be one that produces only conforming basic metafiles (or can be reliably commanded to function in that mode), and additionally conforms to any additional generator requirements as explained in the subsections below.

3.1.2 Conforming basic interpreter. A conforming basic interpreter shall be defined to be one that at least correctly interprets any conforming basic metafile, and conforms to any additional interpreter requirements as explained in the subsections below. In addition, any conforming basic interpreter shall be able to parse and skip any elements that it does not understand or support, and any parameter values that it does not support. For interpreters, there shall be two levels of conformance for judging what comprises "correct" interpretation of a metafile:

Level 1 - Publication Level: All of the specifications of FIPS PUB 128 and of this CGM AP shall be accurately implemented. This includes the guidelines of FIPS PUB 128 annex D.2 and D.5, and the recommendations for the treatment of indeterminate specifications of circular and elliptical primitives in FIPS PUB 128 annex D.4.5. The results shall be completely predictable across implementations conforming at this level; that is, suitable for publication as the name implies.

Level 2 - Draft Level: The guidelines of FIPS PUB 128 annex D.2 (degeneracies) and D.3 (mapping color to black-and-white) shall be implemented. The recommendations for the treatment of indeterminate specifications of circular and elliptical primitives in D.4.5 shall be followed. The capabilities of annex D.5 of FIPS PUB 128 and of the Basic set as defined in this specification shall be present; however, the following interpreter fallback actions of D.4 may be taken:

AUXILIARY COLOR	CHARACTER SPACING
APPEND TEXT	CHARACTER HEIGHT
RESTRICTED TEXT	CHARACTER ORIENTATION
CELL ARRAY	CHARACTER SET INDEX
LINE WIDTH	MARKER SIZE
EDGE WIDTH	TEXT PRECISION
PATTERN SIZE	CHARACTER EXPANSION FACTOR

For LINE TYPE and HATCH INDEX, the interpreter shall have the capabilities of D.5 of FIPS PUB 128. However, the interpreter may take fallback actions for the additional linetypes of Table IV and hatch styles of Table V below.

3.1.3 Limits on parameter data. A conforming basic metafile shall not contain scalar values of parameter data outside the ranges specified by this specification.

3.1.4 Encoding format. A conforming basic metafile shall use only the CGM Binary Encoding, as defined in FIPS PUB 128, part 3. [NOTE: Future CGM application profiles may be developed (or this profile extended) for the character (FIPS PUB 128, part 2) or clear text (FIPS PUB 128, part 4) encodings of CGM.]

3.1.5 Physical file structure. All basic metafiles conforming to this specification shall consist of 80-octet records. When files are being transmitted on magnetic tape, the 80-octet logical records shall be blocked into 800-octet physical records.

3.1.6 Errors in FIPS PUB 128. A number of editorial errors have been found to exist in the published version of ANSI X3.122. In order to prevent errors in the use of FIPS PUB 128 within this specification, the following changes to ANSI X3.122 shall apply:

Part 1, p. 100, the last item on the page: "1" should be "0" and "foreground" should be "background".

Part 3, p.17, item 11: the fraction numerator which is "pnx" should be "pnx-1".

Part 3, p.26, VDC REAL PRECISION: "3I" should be "E,2I".

Part 1, clause 5.2.1 (p. 43), clause 5.3.12 (p. 49), and clause 6 (p. 100): To make clear and remove contradictory statements in these clauses--Metafile Descriptor elements shall not return to default at Begin Picture, and they shall not be included in the Metafile Defaults Replacement.

Part 1, p.106, the expansion of "<metafile contents>": the "|" symbols should be deleted.

3.2 Specific requirements. The following subsections define the specific requirements for a conforming CGM AP. An application profile shall use the specified element types of FIPS PUB 128 with the constraints as specified below.

3.2.1 Metafile constraints. The Basic Set shall be defined by the limitations on Basic Values noted below. Where an element is not mentioned, it is implied that the Basic Set shall include all values permitted in FIPS PUB 128.

3.2.1.1 Delimiter elements. The only constraint on delimiter elements shall be for no-op, and the basic values allowed shall be an arbitrary sequence of n octets, n=0..32767.

3.2.1.2 Metafile descriptor elements. The metafile descriptor element constraints shall be as specified in table I.

TABLE I. Metafile descriptor element constraints

Element	Basic Values
Metafile Description	(Note 1)
Integer Precision	16
Real Precision	(1,16,16) (fixed)
	(0,9,23) (floating point)
Index Precision	16
Colour Precision	8, 16
Colour Index Precision	8, 16
Font List	(Note 2)
Character Set List	(0,4/2) (Note 3)
	(1,4/1) (Note 4)
Character Coding Announcer	0 (Basic 7-bit)
	1 (Basic 8-bit)

Note 1: The Metafile Description element's string: a) shall include a substring briefly identifying company or product, so that interpreters can account for known idiosyncrasies of generators; b) shall contain the substring "MIL-D-28003/BASIC-1".

Note 2: Four simultaneous fonts are supported. The font names are selected from the basic font names in 3.2.5.

Note 3: The character set is ANSI X3.4, 7-bit American National Standard Code for Information Interchange (7-bit ASCII).

Note 4: The character set is ANSI X3.134/2, 8-bit American National Standards Code for Information Interchange (8-bit

ASCII). [**Note:** This is equivalent to ISO 8859/1, Right-Hand Part of Latin Alphabet Number 1.]

3.2.1.3 Picture descriptor elements. Note that the scale-factor parameter of SCALING MODE is always a floating point number, even when REAL PRECISION has selected fixed-point for other real numbers. It is not apparent in FIPS PUB 128 what the precision of this floating point parameter is when fixed point reals have been selected: its precision shall be (0,9,23).

3.2.1.4 Control elements. Control element constraints shall be as specified in table II.

TABLE II. Control element constraints

Element	Basic Values
VDC Integer Precision	16, 32
VDC Real Precision	(1,16,16) (fixed) (0,9,23) (floating point)
Transparency	1 (on)

Note: VDC stands for Virtual Device Coordinates, the coordinate system of FIPS PUB 128.

3.2.1.5 Graphical primitives. To ensure portability and predictable results, conforming basic metafiles shall not contain any Generalized Drawing Primitive (GDP) elements. [**Note:** Future addenda to this specification may specify GDP elements to be included in the Basic set.]

3.2.1.6 Attribute elements. Attribute element constraints shall be as specified in table III.

TABLE III. Attribute element constraints

Element	Basic Values
Line Bundle Index	1-5
Line Type	1-5 (Note 1)
Marker Bundle Index	1-5
Marker Type	1-5
Text Bundle Index	1-2
Text Font Index	1-4
Character Set Index	1-2
Alternate Character Set Index	1-2
Fill Bundle Index	1-5
Hatch Index	1-6 (Note 2)
Edge Bundle Index	1-5
Edge Type	1-5
Pattern Table	Starting Index, 1-8 nx, 1-16 ny, 1-16
Color Table	start index 0-255

Note 1: Additionally, the linetypes defined in 3.2.2.1 shall be included in the Basic Set of this specification.

Note 2: Additionally, the hatch styles (indexes) defined in 3.2.2.2 shall be included in the Basic Set of this specification.

For indexed color selection, either all color indexes used in the metafile shall have their representations defined by use of the COLOR TABLE element, or none shall. A color index is "used" if it occurs in an element selecting a color value to be applied to a primitive (LINE COLOR, CELL ARRAY, etc).

3.2.1.7 Escape element. To ensure portability and predictable results, CGM application profiles conforming to this specification may contain only those ESCAPE elements that are defined in 3.2.6.

3.2.1.8 External elements. The "action required" flag of the MESSAGE element shall be restricted to the value "no action required."

3.2.2 Additional attribute values

3.2.2.1 Linetypes. The additional linetypes specified in table IV shall apply (see figures 1 through 10 for examples).

TABLE IV. Additional linetypes

Linetype	CGM parameter value
chain line	-11301
center line	-11302
hidden line	-11303
phantom line	-11304
double arrow	-11305
single dot	-11306
single arrow	-11307
stitch line	-11308
break line, style 1	-11309
break line, style 2	-11310

3.2.2.2 Hatch styles. The additional hatch styles specified in table V shall apply (see figure 11 for examples).

TABLE V. Additional hatch styles

Hatch style	CGM parameter value
across grain wood	-11401
with grain wood	-11402
bronze, brass, copper, and compositions	-11403
cast iron or malleable iron and general use for all materials	-11404
steel	-11405
concrete	-11406
cork, felt, fabric, leather, and fiber	-11407
magnesium, aluminum, and aluminum alloys	-11409
marble, slate, glass, porcelain, etc.	-11410
rock	-11411
rubber, plastic, and electrical insulation	-11412
sand	-11413
sound insulation	-11414
thermal insulation	-11415
titanium and refractory material	-11416
water and other liquids	-11417
white metal, zinc, lead, babbitt, and alloys	-11418

3.2.3 FIPS PUB 128 defaults. FIPS PUB 128 specifies a complete set of defaults. In some cases, these defaults do not match the application requirements of this specification. However, any conforming basic metafile satisfying this specification shall be a legal CGM as specified in FIPS PUB 128, including implicit

defaults. Thus, each deviation requires that the affected element shall either:

1. Appear in the METAFILE DEFAULTS REPLACEMENT element, or

2. Be explicitly specified for its value to be applicable.

Therefore, any conforming basic metafile satisfying this specification shall contain in the Metafile Descriptor a METAFILE DEFAULTS REPLACEMENT element that includes (at a minimum):

TEXT PRECISION element; precision 2 (stroke).

3.2.4 Specification of semantic ambiguities. FIPS PUB 128 leaves the semantics of a number of graphical details unspecified or "implementation dependent." This is unacceptable where predictable interchange is required. The following specifications shall apply for conforming basic generators and interpreters of this specification:

3.2.4.1 View surface clearing. Unless clearing is suppressed by 'Escape -301', the view surface shall be cleared upon interpretation of the Begin Picture Body element.

3.2.4.2 Clipping. When the clip indicator is "off", clipping shall be done to the intersection of the device viewport and the device view surface limits. When clipping is "on", clipping shall be done to the intersection of the clip rectangle, the VDC extent, the device viewport and the device view surface limits.

3.2.4.3 Linetype continuation. Linetype shall be maintained (continued) across the interior vertices of a polyline.

3.2.4.4 Edge type continuation. Edge type shall be maintained (continued) across the vertices of a filled area boundary.

3.2.5 Font specifications. The fonts in table VI are public domain fonts, available as part of NBS SP 424. All of these fonts shall be considered basic capabilities of a basic metafile conforming to this specification. Any of these fonts may appear in the Font List element in a basic metafile that conforms to this specification. Font name shall be the concatenation of the string "HERSHEY:", to designate one of the Hershey fonts, and a "name string" to designate the particular typeface.

TABLE VI. Basic font names

1.	HERSHEY:CARTOGRAPHIC_ROMAN
2.	HERSHEY:CARTOGRAPHIC_GREEK
3.	HERSHEY:SIMPLEX_ROMAN
4.	HERSHEY:SIMPLEX_GREEK
5.	HERSHEY:SIMPLEX_SCRIPT
6.	HERSHEY:COMPLEX_ROMAN
7.	HERSHEY:COMPLEX_GREEK
8.	HERSHEY:COMPLEX_SCRIPT
9.	HERSHEY:COMPLEX_ITALIC
10.	HERSHEY:COMPLEX_CYRILLIC
11.	HERSHEY:DUPLEX_ROMAN
12.	HERSHEY:TRIPLEX_ROMAN
13.	HERSHEY:TRIPLEX_ITALIC
14.	HERSHEY:GOTHIC_GERMAN
15.	HERSHEY:GOTHIC_ENGLISH
16.	HERSHEY:GOTHIC_ITALIA

3.2.6 Escape elements. Support of the following ESCAPE elements shall be required in conforming implementations under this specification.

3.2.6.1 Disable clearing of view surface. The normal interpretation of a CGM metafile is such that the view surface of a device is cleared on each Begin Picture Body element. This Escape element will disable the clearing of the view surface for all of the pictures in the metafile. The effect of this Escape element is to permit multiple metafile pictures to be imaged on the same view surface with a mapping as described in FIPS PUB 128. The pictures may have different VDC Extents. Thus, each picture shall be mapped into the current device viewport (whether default or specified by the Device Viewport Escape element). If used, this Escape element must appear in the Metafile Descriptor. This Escape element shall be a basic capability of the CGM Application Profile under this specification.

Escape Identifier: -301

Escape Data Record: null

3.2.6.2 Device viewport. The default device viewport is the largest rectangular area of the screen. The viewport shall be redefined by this escape element by specifying two corner points. The effective viewport shall be the largest rectangle in the viewport having the same aspect ratio as the VDC extent, and having the same lower-left corner as the device viewport. The VDC extent shall be mapped onto the effective viewport when the

picture is interpreted. The units with which the two points are specified shall be real fractions [0.0 to 1.0], which shall be applied to the device viewport. If used, this Escape must appear in the Picture Descriptor. If the scaling mode has been set to metric, then the device viewport shall have precedence -- the scaling mode shall be treated as if it were abstract.

Escape Identifier: -302

Escape Data Record: A single string of text containing the specification of the viewport. Parameters in the viewport shall be separated by at least one blank character and/or a single comma character. The decimal point of the real fraction shall be required. Leading zeroes of the real fraction shall be optional. There are four parameters:

P1: First corner x-coordinate. Real fraction of the default device viewport, in the range [0.0,1.0].

P2: First corner y-coordinate. Real fraction of the default device viewport, in the range [0.0,1.0].

P3: Second corner x-coordinate. Real fraction of the default device viewport, in the range [0.0,1.0].

P4: Second corner y-coordinate. Real fraction of the default device viewport, in the range [0.0,1.0].

Example: a viewport equal to the upper right quarter of the default viewport could be coded as:

Escape Identifier -302

Escape Data Record ".5 .5 1. 1."

This Escape element shall be a basic capability of the CGM Application Profile of this specification.

3.2.6.3 Implicit Color Table. The default Color Table is undefined in FIPS PUB 128. It is always possible to specify an entire default Color Table with Metafile Defaults Replacement. This ESCAPE element shall provide a shorthand method to specify a default color table from a small set of predefined tables, and

shall provide a method to specify how the interpreter shall define its own color table according to available resources. This ESCAPE shall be allowed in the Metafile Descriptor. If used, no COLOR TABLE elements shall appear in the metafile. The single integer parameter of the ESCAPE shall have the following meanings:

0: "none" -- there shall be no implicit default value of the color table, interpreters shall associate representations with color indexes as they see fit.

1: "cyclic" -- the interpreter shall initialize its color table to contain

0 - white
1 - black
2 - red
3 - green
4 - blue
5 - yellow
6 - magenta
7 - cyan
8 - black
9 - white
10 through 255 - cyclic repetition of 2-9.

2: "uniform" -- the interpreter shall initialize its color table to: black and white in indexes 0 and 1. Beginning at index 2, 224 representations that shall uniformly sample the RGB color cube as follows:

- 5 levels of red evenly spaced from none to maximum;
- 9 levels of green evenly spaced from none to maximum;
- 5 levels of blue evenly spaced from none to maximum.

The color cube shall be mapped to the linear array by varying the red dimension most rapidly, then the green dimension, then the blue dimension. Note that the 225th element implied by this sequence, black, shall not be put into the table. Beginning at index 226 shall be 28 levels of gray. Index 226 shall be 1/32 (very dark). Succeeding entries, with the exceptions described below, shall be 1/32 lighter. The exceptions shall be 0/32, 8/32, 16/32, 24/32, 32/32, which shall be found in the "color cube" section of the table.

The default value shall be 1, "cyclic."

Escape Identifier: -303

Escape Data Record: A single string of text containing the single integer parameter of this escape element. The integer is encoded as "clear text", i.e., value 2 is encoded as the string comprised of (or containing) the ASCII character "2".

This Escape element shall be a basic capability of the CGM Application Profile of this specification.

3.2.7 Implementation dependencies. This section specifies implementation dependencies and environmental constraints for CGM APs conforming to this specification.

3.2.7.1 General guidelines for FIPS PUB 128 elements. Unless otherwise noted in this specification, the guidelines of FIPS PUB 128 Annex D shall apply to conforming basic generators and interpreters as defined in 3.1.

Name: Metafile Defaults Replacement

Description: The Metafile Defaults Replacement element shall not be partitioned. Note that FIPS PUB 128 permits multiple occurrences of this element, so that partitioning is not required. Partitioning shall be permitted for all other elements.

Name: Restricted Text

Description: Draft level capability of a basic conforming interpreter shall be to render the complete restricted text string (including appended text), scaled isotropically (i.e., specified aspect ratio for the text is not distorted) such that the string fits into the text extent parallelogram.

Name: Color Table

Description: The Color Table element has an unspecified effect when it appears in a picture subsequent to any graphical primitives. If a Color Table element defining the representation of a given color index appears in a picture, it shall appear before reference to that index by an attribute

element or use of that index by a graphical primitive element (included in the latter shall be implicit use of default color index attribute values by the first occurrence of an associated primitive). Once a given color representation is defined and used, it shall not be redefined. [**Note:** These restrictions insure that interpreting systems without dynamic color update capabilities shall be able to render the intended picture accurately.]

Name: Pattern Table

Description: The Pattern Table element has an unspecified effect when it appears in a picture subsequent to any graphical primitives filled with the affected pattern index. If a Pattern Table element defining the representation of a given pattern index appears in a picture: a) it shall appear before explicit reference to that index by any Pattern Index element; or b) in the case of the default pattern index, it shall appear before any implicit reference caused by the first occurrence of an associated filled primitive. Once a given pattern representation is defined and used, it shall not be redefined. [**Note:** These restrictions insure that interpreting systems without dynamic pattern update capabilities shall be able to render the intended picture accurately.]

3.2.8 Implementation requirements for conforming basic generators and interpreters. The specifications in this section shall augment those of FIPS PUB 128, Part 1, annex D.5, and Part 3, clause 8.

3.2.8.1 Additional generator specifications. This specification states that the values of attributes (e.g., linetype) shall be restricted to a certain set. When a conforming basic generator receives (from the application or graphics system client) a value outside of the Basic set, it shall be handled as follows:

If the index is selecting an attribute (e.g., linetype), then a conforming basic generator shall map it to the default value for that attribute;

If the index is **defining** an attribute (e.g., color table), then it shall be ignored if outside the Basic range.

These choices for a conforming basic generator are consistent with Part 1, annex D of FIPS PUB 128.

Conforming generators shall provide to applications the means to either select the implicit color table for a metafile (e.g., via a mechanism to cause generation of 'Escape -303') or to ascertain the value that will be used in the metafile. The means to ascertain the value shall consist of either a software inquiry mechanism or documentation accompanying the system. An inquiry mechanism shall be the preferred method.

3.2.8.2 Additional basic interpreter specifications. In the absence of any color table elements, or of ESCAPE -303 (see 3.2.6.3), in the metafile, conforming basic interpreters shall initialize their color tables as follows: index 0 shall be set to white; index 1 shall be set to black; and indexes 2-254 shall be set by cyclic repetition of the 8 entries specified in table VII. Draft Level conformance for interpreters shall allow black and white to be reversed in the first two indices.

TABLE VII. Default Color Table

Index	Values	Meaning
2	(1.0,0,0)	Red
3	(0,1.0,0)	Green
4	(0,0,1.0)	Blue
5	(1.0,1.0,0)	Yellow
6	(1.0,0,1.0)	Magenta
7	(0,1.0,1.0)	Cyan
8	(0,0,0)	Black
9	(1.0,1.0,1.0)	White

Note: The values '1.0' in the preceding table denote full intensity for the appropriate component.

3.2.8.3 Minimum data structure support. The following named elements shall have basic values as defined below:

Name: Maximum Color Array Dimension

Description: The basic value for the number of color values that can appear in a color array or color list parameter shall be: 1048576 for CELL ARRAY (one 1024x1024 image); 2048 for

PATTERN TABLE (eight 16x16 patterns); 256 for COLOR TABLE (entries 0-255). CELL ARRAY and PATTERN TABLE have color array parameters and COLOR TABLE has a color list parameter.

Name: Maximum Point Array Length

Description: The basic value for the number of points and VDC that can appear in parameters for metafile elements shall be 1024.

Name: Maximum String Length

Description: The basic value for the length of an individual string of characters shall be: 254 for all string parameters except data records; 32767 for data records.

Name: Bundle Table

Description: Bundle representations are not settable in the current version of FIPS PUB 128. To insure predictable results, interpreters and generators conforming to the CGM Application Profile of this specification shall use the default values from table VIII.

TABLE VIII. Default bundle tables

	Bundle Index				
Bundle Type	1	2	3	4	5
Line Bundle					
Line Type	solid	dash	dot	dash-dot	dash-dot-dot
Line Width	1	1	1	1	1
Line Color	1	1	1	1	1
Marker Bundle					
Marker Type	dot	plus	asterisk	circle	cross
Marker Size	1	1	1	1	1
Marker Color	1	1	1	1	1
Text Bundle					
Font Index	1	1			
Text Precision	stroke	stroke			
Character Expansion Factor	1	0.7			
Char. Spacing	0	0			
Text Color	1	1			
Fill Bundle					
Interior Style	hatch	hatch	hatch	hatch	hatch
Fill Color	1	1	1	1	1
Hatch Index	1	2	3	4	5
Pattern Index	1	1	1	1	1
Edge Bundle					
Edge Type	solid	dash	dot	dash-dot	dash-dot-dot
Edge Width	1	1	1	1	1
Edge Color	1	1	1	1	1

4. QUALITY ASSURANCE PROVISIONS

4.1 Responsibility for inspection. Unless otherwise specified in the contract or purchase order, the contractor is responsible for the performance of all inspection requirements (examinations and tests) as specified herein. Except as otherwise specified in the contract or purchase order, the contractor may use his own or any other facilities suitable for the performance of the inspection requirements herein, unless disapproved by the Government. The Government reserves the right to perform any of the inspections set forth in the specification where such inspections are deemed necessary to ensure that supplies and services conform to prescribed requirements.

4.2 Responsibility for compliance. All items shall meet all requirements of section 3. The inspection set forth in this specification shall become a part of the contractor's overall inspection system or quality program. The absence of any inspection requirements in the specification shall not relieve the contractor of the responsibility of ensuring that all products or supplies submitted to the Government for acceptance comply with all requirements of the contract. Sampling in quality conformance does not authorize submission of known defective material, either indicated or actual, nor does it commit the Government to acceptance of defective material.

4.3 Inspection procedures. All entities, attributes and parameter values shall be analyzed for conformance to FIPS PUB 128 and to section 3 of this specification for a conforming basic metafile. This shall be accomplished with an appropriate software utility, or conformance test suite. All conforming basic metafiles contained in a particular CGM application profile shall be displayed and checked visually for conformance to the requirements of FIPS PUB 128 and of section 3 in its entirety.

4.3.1 Font rendering. Font names shall be specified in a manner compatible with ISO DIS 9541, Font and Character Information Interchange, when it becomes stable. Until then, this specification shall consider any rendering of a requested font conforming if the rendering is "metrically identical" to the font metrics of the requested font. This means that the placement and alignment of the string and the placement, size, and shape of individual characters (i.e., the drawn portions of the character cells) shall be measurably identical. This does allow a good quality filled font to be substituted for a stroked Hershey font, for example. Finally, the Hershey "fonts" are really a mixture of fonts and character sets (e.g., Greek is a character set). The requirements of this specification shall be served by

providing that the necessary character sets be supported in part, and the necessary typefaces be supported in part, so that the combinations required to render the listed 16 Hershey "fonts" shall be supported in full. It is recognized that the Hershey fonts may not be of adequate quality for modern publication requirements.

4.3.2 Error processing. A conforming CGM interpreter shall gracefully recover from any exception condition. If there is something which is not understood by the interpreter, then if possible that element should be skipped, appropriate error warnings generated or logged, and interpretation continue with the next element following the problem element.

5. PACKAGING

Packaging of illustration data files for delivery shall be in accordance with the requirements of MIL-STD-1840.

6. NOTES

6.1 Intended use. This specification is designed to be incorporated into a contract to define the technical requirements to be met when it is desired to purchase illustration or picture description data (in contrast to product definition data) in digital form for use in technical illustrations and technical publications. A CGM AP under this specification represents illustration data in the form of a conforming basic metafile, i.e., it contains, in device-, system-, and implementation-independent form, the picture description data represented by the functions invoked through an application program interface. A CGM AP contains the allowable output primitives and attributes which may be used to compose the picture. In addition, the CGM AP of this specification specifies certain constraints on CGM generators and interpreters to remove implementation dependencies, thereby serving to ensure predictable interchange of conforming basic metafiles between clients.

6.1.1 Explanation of CGM AP. The syntactic specification in the FIPS PUB 128 is complete and unambiguous. It is, as well, redundant in the sense that there are three distinct encodings of the same functionality: binary, character, and clear text. The redundancy serves a useful purpose, as each encoding is tailored to certain computing environments and applications, and so the CGM client has the opportunity to choose a syntax that is optimized to the intended application. The binary encoding has been chosen as the only encoding which will be supported by this military specification at this time.

The semantic specification is less complete. The expected overall results of using the geometric primitive elements are well enough specified. However some of the finer details, such as the precise appearance of joints and endpoints in lines, are unspecified. This underspecification of semantics was intentional on the part of the standards committees formulating the CGM standard, since it allows a wider range of existing systems to be accommodated and makes the standard more adaptable to the various needs and philosophies of a diverse clientele.

On the other hand, the semantic ambiguity does mean that there will be no single correct interpretation of a given CGM metafile, and hence it will be difficult to unambiguously describe an

intended picture using the CGM standard. This is a distinct drawback in certain application environments, such as the areas of Technical Illustration and Technical Publishing.

There are further sources of uncertainty in using CGM in an application environment. A CGM metafile is produced by a component of a graphics environment known as a "metafile generator." The content of a CGM metafile is rendered into pictures by a component known as a "metafile interpreter." FIPS PUB 128 specifically excludes standardization of the behavior of metafile generators and metafile interpreters. (Most such behavior is described as "implementation dependent.") In doing so, a certain unpredictability of results is introduced into the graphics system viewed as a whole; for example, CGM generators serving GKS (Graphical Kernel System, ANSI X3.124) clients in the product lines of two different vendors might map out-of-range attributes differently.

These two sources of ambiguity in using the CGM standard--incomplete semantics and non-specification of the behavior of generators and interpreters--do not diminish the utility of FIPS PUB 128 for technical illustration and technical publishing. It is a sound and suitable basic protocol for these areas. But they do mean that some further specification (beyond that in the published standard) is required in order for the use of the CGM standard to be effective and unambiguous.

Such a specification is precisely what an Application Profile (AP) consists of. In the case of CGM, an AP specifies:

1. complete semantics;

2. the behavior of CGM generators and CGM interpreters;

An AP specifies minimal and maximal requirements for generators and interpreters, and ties down all implementation dependencies of the CGM metafile. As the name suggests, the AP for CGM is a set of specifications appropriate to a given application environment.

6.1.2 <u>Metafile Descriptor Elements</u>. It is unclear in FIPS PUB 128 whether there should be a mandatory ordering of Metafile Descriptor elements (the grammar implies some). Addendum 1 of FIPS PUB 128 will impose such an ordering when it becomes part of the standard. This is to point out that such an ordering may become mandatory in a future revision of this specification.

6.1.3 Additional attribute values.

6.1.3.1 Linetypes. The linetypes specified in table IV of 3.2.2.1 have been submitted to ANSI and ISO, the International Standards Organization, for graphics registration (see figures 1 through 10 for examples). The figures for these linetypes are taken from the latest draft of the registration proposals that have been submitted to ANSI and ISO. In table IV, the name of the linetype is given, followed by the numeric value (the linetype parameter) by which it is to be referenced. These references may change in future amendments to this specification.

6.1.3.2 Hatch styles. The hatch styles in table V of 3.2.2.2 have also been submitted for graphical registration (see figure 11 for examples). The hatch style examples of figure 11 are taken from ANSI Y14.26, Engineering Drawing and Related Documentation Practices--Digital Representation for Communication of Product Definition Data. In table V, the name of the hatch style is given, followed by the numeric value (the hatch index parameter) by which it is to be referenced. This reference may change in future amendments to this specification.

6.2 Ordering data. The contract or purchase order should specify the following:

a. Title, number, and date of this specification.

b. The application profile should specify whether it is meant for publication level or draft level interpretation. (See 3.1.2)

6.3 Definitions.

6.3.1 Acronyms and abbreviations used in this specification. Acronyms and abbreviations used in this specification are defined as follows:

a. ANSI - The American National Standards Institute.

b. AP - Application Profile.

c. CGM - Computer Graphics Metafile. Synonymous with FIPS PUB 128.

d. DIS - Draft International Standard.

e. FIPS - Federal Information Processing Standards.

f. GDP - Generalized Drawing Primitive.

g. GKS - Graphical Kernel System.

h. ISO - International Standards Organization.

i. PUB - Publication.

j. SP - Special Publication.

k. VDC - Virtual Device Coordinates, the coordinate system of FIPS PUB 128.

6.3.2 Application Profile. A specification that defines the use of a standard, and defines all possible data streams that conform to that profile. An AP insures interoperability of different/multiple implementations of a standard. In this context, it completely and unambiguously represents the information requirements for a particular application of digital graphics data.

6.3.3 Basic values. The subset of permissible values for parameters of a CGM element that are mandatory for conformance to this specification.

6.3.4 Computer Graphics Metafile. The specification for a mechanism for storing and transferring illustration data. Refer to FIPS PUB 128.

6.3.5 Conforming basic generator. A metafile generator that produces only conforming basic metafiles (or can be reliably commanded to function in that mode), and additionally conforms to any additional generator requirements as explained in section 3.

6.3.6 Conforming basic interpreter. An metafile interpreter that at least correctly interprets any conforming basic metafile, and conforms to any additional interpreter requirements as explained in section 3.

6.3.7 Draft level. The metafile interpreter level for all documents except those for final document production.

6.3.8 Metafile. Synonymous with CGM. A representation for the storage and transfer of graphical data and control information. This representation contains a device-independent description of one or more pictures.

6.3.9 Metafile generator. The software or hardware that creates a picture or conveys information in the CGM representation.

6.3.10 **Metafile interpreter**. The software or hardware that reads a CGM metafile and interprets the contents.

6.3.11 **Permissible values**. The range of values for a parameter of a CGM element as specified in FIPS PUB 128.

6.3.12 **Publication level**. The metafile interpreter level for final document production.

6.3.13 **Vector Graphics**. The presentation or storage of images as sequences of line segments.

Note: Refer to FIPS PUB 128, clause 3, for further definitions of computer graphics terms.

6.4 **Subject term (keyword) listing**.

Application profile
CGM
CGM metafile
Digital
FIPS PUB 128
Technical illustrations
Technical publications

Proposal Number:	2
Presentation date of proposal:	10 April 1987
Sponsoring Authority:	ANSI
Class of Graphical Item:	LINETYPE
Name:	single arrow

Description

A single arrow linetype consists of a solid line terminated by an arrowhead as specified in ANSI Y14.2M-1979 (Line Conventions and Lettering) requirements for dimension and leader lines. The arrow is rendered so that the arrow tip occurs at the last point of the list of points passed to a polyline and is in the direction of the last vector.

This linetype is intended for use in engineering drawings.

Specific details are implementation dependent. The appearance of the degenerate case is implementation dependent.

Such a linetype has the following visual appearance:

Additional Comments

The requirements stated in ANSI Y14.2M-1979 shall be followed when rendering this linetype.

This linetype is not intended to be used as an edgetype.

Justification for Inclusion

This linetype is commonly used in engineering drawings. It is one of a set of linetypes to be registered for use with computer graphics standards to enable compact storage and transfer of engineering drawings.

Relationship to Standards

1) ISO 7942 (GKS) - Specifies a registered linetype to supplement those defined in 5.4.1.
2) ISO 8632 (CGM) - Specifies a registered linetype to supplement those defined in 5.7.2.
3) ANSI Y14.2M-1979 - Line Conventions and Lettering.

FIGURE 1: Example of linetype single arrow.

Proposal Number:	3
Presentation date of proposal:	10 April 1987
Sponsoring Authority:	ANSI
Class of Graphical Item:	LINETYPE
Name:	single dot

Description

A single dot linetype consists of a solid line terminated by a dot as specified in ANSI Y14.2M-1979 (Line Conventions and Lettering) requirements for leader lines. The dot is rendered so that the dot occurs at the last point in the list of points passed to a polyline.

This linetype is intended for use in engineering drawings.

Specific details are implementation dependent. The appearance of the degenerate case is implementation dependent.

Such a linetype has the following visual appearance:

Additional Comments

The requirements stated in ANSI Y14.2M-1979 shall be followed when rendering this linetype.

This linetype is not intended to be used as an edgetype.

Justification for Inclusion

This linetype is commonly used in engineering drawings. It is one of a set of linetypes registered for use with computer graphics standards to enable compact storage and transfer of engineering drawings.

Relationship to Standards

1) ISO 7942 (GKS) - Specifies a registered linetype to supplement those defined in 5.4.1.
2) ISO 8632 (CGM) - Specifies a registered linetype to supplement those defined in 5.7.2.
3) ANSI Y14.2M-1979 - Line Conventions and Lettering.

FIGURE 2: Example of linetype single dot.

Proposal Number:	4
Presentation date of proposal:	10 April 1987
Sponsoring Authority:	ANSI
Class of Graphical Item:	LINETYPE
Name:	double arrow

Description

A double arrow linetype consists of a solid line terminated by two arrowheads as specified in ANSI Y14.2M-1979 (Line Conventions and Lettering) requirements for dimension lines. The arrows are rendered so that the arrow tip occurs at the first and last points in the list of points passed to a polyline.

This linetype is intended for use in engineering drawings.

Specific details are implementation dependent. The appearance of the degenerate case is implementation dependent.

Such a linetype has the following visual appearance:

Additional Comments

The requirements stated in ANSI Y14.2M-1979 shall be followed when rendering this linetype.

This linetype is not intended to be used as an edgetype.

Justification for Inclusion

This linetype is commonly used in engineering drawings. It is one of a set of linetypes registered for use with computer graphics standards to enable compact storage and transfer of engineering drawings.

Relationship to Standards

1) ISO 7942 (GKS) - Specifies a registered linetype to supplement those defined in 5.4.1.
2) ISO 8632 (CGM) - Specifies a registered linetype to supplement those defined in 5.7.2.
3) ANSI Y14.2M-1979 - Line Conventions and Lettering.

FIGURE 3: Example of linetype double arrow.

Proposal Number:	5
Presentation date of proposal:	10 April 1987
Sponsoring Authority:	ANSI
Class of Graphical Item:	LINETYPE
Name:	stitch line

Description

A stitch line linetype consists of dashes and spaces of equal length as specified in ANSI Y14.2M-1979 (Line Conventions and Lettering.)

This linetype is intended for use in engineering drawings. Its definition contains rendition requirements beyond those for the dashed linetype already present in the graphics standards.

Specific details are implementation dependent. The appearance of the degenerate case is implementation dependent.

Such a line has the following visual appearance:

Additional Comments

The requirements stated in ANSI Y14.2M-1979 shall be followed when rendering this linetype. In some cases, it is necessary for the standard (e.g. GKS) to exercise precise control over the manner in which two lines intersect in a drawing. In these cases it may be appropriate for the client to simulate this linetype by using sequences of correctly placed individual line segments.

This linetype is not intended to be used as an edgetype.

Justification for Inclusion

This linetype is commonly used in engineering drawings. It is one of a set of linetypes registered for use with computer graphics standards to enable compact storage and transfer of engineering drawings.

Relationship to Standards

1) ISO 7942 (GKS) - Specifies a registered linetype to supplement those defined in 5.4.1.
2) ISO 8632 (CGM) - Specifies a registered linetype to supplement those defined in 5.7.2.
3) ANSI Y14.2M-1979 - Line Conventions and Lettering.

FIGURE 4: Example of linetype stitch line

Proposal Number:	6
Presentation date of proposal:	10 April 1987
Sponsoring Authority:	ANSI
Class of Graphical Item:	LINETYPE
Name:	chain line

Description

A chain line linetype consists of alternating long and short dashes as specified in ANSI Y14.2M-1979 (Line Conventions and Lettering.)

This linetype is intended for use in engineering drawings. Its rendition is different from that of the dashed-dotted linestyle already present in the graphics standards.

Specific details are implementation dependent. The appearance of the degenerate case is implementation dependent.

Such a line has the following visual appearance:

Additional Comments

The requirements stated in ANSI Y14.2M-1979 shall be followed when rendering this linetype. In some cases, it is necessary for the standard (e.g. GKS) to exercise precise control over the manner in which two lines intersect in a drawing. In these cases it may be appropriate for the client to simulate this linetype by using sequences of correctly placed individual line segments.

This linetype is not intended to be used as an edgetype.

Justification for Inclusion

This linetype is commonly used in engineering drawings. It is one of a set of linetypes registered for use with computer graphics standards to enable compact storage and transfer of engineering drawings.

Relationship to Standards

1) ISO 7942 (GKS) - Specifies a registered linetype to supplement those defined in 5.4.1.
2) ISO 8632 (CGM) - Specifies a registered linetype to supplement those defined in 5.7.2.
3) ANSI Y14.2M-1979 - Line Conventions and Lettering.

FIGURE 5: Example of linetype chain line.

Proposal Number:	7
Presentation date of proposal:	10 April 1987
Sponsoring Authority:	ANSI
Class of Graphical Item:	LINETYPE
Name:	center line

Description

A center line linetype consists of alternating long and short dashes as specified in ANSI Y14.2M-1979 (Line Conventions and Lettering.)

This linetype is intended for use in engineering drawings. The long dashes may vary in length depending on the size of the drawing. Lines drawn in this linetype shall start and end with long dashes. A very short line may be unbroken.

Specific details are implementation dependent. The appearance of the degenerate case is implementation dependent.

Such a line has the following visual appearance:

Additional Comments

The requirements stated in ANSI Y14.2M-1979 shall be followed when rendering this linetype. In some cases, it is necessary for the standard (e.g. GKS) to exercise precise control over the manner in which two center lines intersect in a drawing. In these cases it may be appropriate for the client to simulate this linetype by using sequences of correctly placed individual line segments.

This linetype is not intended to be used as an edgetype.

Justification for Inclusion

This linetype is commonly used in engineering drawings. It is one of a set of linetypes registered for use with computer graphics standards to enable compact storage and transfer of engineering drawings.

Relationship to Standards

1) ISO 7942 (GKS) - Specifies a registered linetype to supplement those defined in 5.4.1.
2) ISO 8632 (CGM) - Specifies a registered linetype to supplement those defined in 5.7.2.
3) ANSI Y14.2M-1979 - Line Conventions and Lettering.

FIGURE 6: Example of linetype center line

Proposal Number: 8

Presentation date of proposal: 10 April 1987

Sponsoring Authority: ANSI

Class of Graphical Item: LINETYPE

Name: hidden line

Description

A hidden line linetype consists of short evenly spaced dashes as specified in ANSI Y14.2M-1979 (Line Conventions and Lettering.)
This linetype is intended for use in engineering drawings. The dashes may vary in length depending on the size of the drawing. Lines, not polylines, drawn in this linetype shall start and end with a dash. Dashes will join at corners, and arcs drawn with this style shall start and end with dashes. These rendition requirements are different from the dashed linetype already defined in the graphics standards.

Specific details are implementation dependent. The appearance of the degenerate case is implementation dependent.

Such a line has the following visual appearance:

Additional Comments

The requirements stated in ANSI Y14.2M-1979 shall be followed when rendering this linetype. In some cases, it is necessary for the standard (e.g. GKS) to exercise precise control over the manner in which two lines intersect in a drawing. In these cases it may be appropriate for the client to simulate this linetype by using sequences of correctly placed individual line segments.

This linetype is not intended to be used as an edgetype.

Justification for Inclusion

This linetype is commonly used in engineering drawings. It is one of a set of linetypes registered for use with computer graphics standards to enable compact storage and transfer of engineering drawings.

Relationship to Standards

1) ISO 7942 (GKS) - Specifies a registered linetype to supplement those defined in 5.4.1.
2) ISO 8632 (CGM) - Specifies a registered linetype to supplement those defined in 5.7.2.
3) ANSI Y14.2M-1979 - Line Conventions and Lettering.

FIGURE 7: Example of linetype hidden line

Proposal Number: 9

Presentation date of proposal: 10 April 1987

Sponsoring Authority: ANSI

Class of Graphical Item: LINETYPE

Name: phantom line

Description

A phantom line linetype consists of long dashes separated by pairs of short dashes as specified in ANSI Y14.2M-1979 (Line Conventions and Lettering.)

This linetype is intended for use in engineering drawings. Lines, not polylines, drawn in this linetype shall start and end with long dashes which may vary in length depending on the size of the drawing.

Specific details are implementation dependent. The appearance of the degenerate case is implementation dependent.

Such a line has the following visual appearance:

Additional Comments

The requirements stated in ANSI Y14.2M-1979 shall be followed when rendering this linetype. In some cases, it is necessary for the standard (e.g. GKS) to exercise precise control over the manner in which two lines intersect in a drawing. In these cases it may be appropriate for the client to simulate this linetype by using sequences of correctly placed individual line segments.

This linetype is not intended to be used as an edgetype.

Justification for Inclusion

This linetype is commonly used in engineering drawings. It is one of a set of linetypes registered for use with computer graphics standards to enable compact storage and transfer of engineering drawings.

Relationship to Standards

1) ISO 7942 (GKS) - Specifies a registered linetype to supplement those defined in 5.4.1.
2) ISO 8632 (CGM) - Specifies a registered linetype to supplement those defined in 5.7.2.
3) ANSI Y14.2M-1979 - Line Conventions and Lettering.

FIGURE 8: Example of linetype phantom line.

Proposal Number: 10

Presentation date of proposal: 10 April 1987

Sponsoring Authority: ANSI

Class of Graphical Item: LINETYPE

Name: break line - style 1

Description

A break line linetype - style 1 - consists of either one of two allowable representations as specified in ANSI Y14.2M-1979 (Line Conventions and Lettering.) This is simply a line having a "freehand" appearance.

This linetype is intended for use in engineering drawings.

Specific details are implementation dependent. The appearance of the degenerate case is implementation dependent.

Such a line has the following visual appearance:

Additional Comments

The requirements stated in ANSI Y14.2M-1979 shall be followed when rendering this linetype.

This linetype is not intended to be used as an edgetype.

Justification for Inclusion

This linetype is commonly used in engineering drawings. It is one of a set of linetypes registered for use with computer graphics standards to enable compact storage and transfer of engineering drawings.

Relationship to Standards

1) ISO 7942 (GKS) - Specifies a registered linetype to supplement those defined in 5.4.1.
2) ISO 8632 (CGM) - Specifies a registered linetype to supplement those defined in 5.7.2.
3) ANSI Y14.2M-1979 - Line Conventions and Lettering.

FIGURE 9: Example of linetype break line, style 1

Proposal Number:	11
Presentation date of proposal:	10 April 1987
Sponsoring Authority:	ANSI
Class of Graphical Item:	LINETYPE
Name:	break line - style 2

Description

A break line linestyle consists of either one of two allowable representations as specified in ANSI Y14.2M-1979 (Line conventions and Lettering.) This is a line consisting of long dashes joined by zigzags.

This linetype is intended for use in engineering drawings.

Specific details are implementation dependent. The appearance of the degenerate case is implementation dependent.

Such a line has the following visual appearance:

Additional Comments

The requirements stated in ANSI Y14.2M-1979 shall be followed when rendering this linetype.

This linetype is not intended to be used as an edgetype.

Justification for Inclusion

This linetype is commonly used in engineering drawings. It is one of a set of linetypes registered for use with computer graphics standards to enable compact storage and transfer of engineering drawings.

Relationship to Standards

1) ISO 7942 (GKS) - Specifies a registered linetype to supplement those defined in 5.4.1.
2) ISO 8632 (CGM) - Specifies a registered linetype to supplement those defined in 5.7.2.
3) ANSI Y14.2M-1979 - Line Conventions and Lettering.

FIGURE 10: Example of linetype break line, style 2.

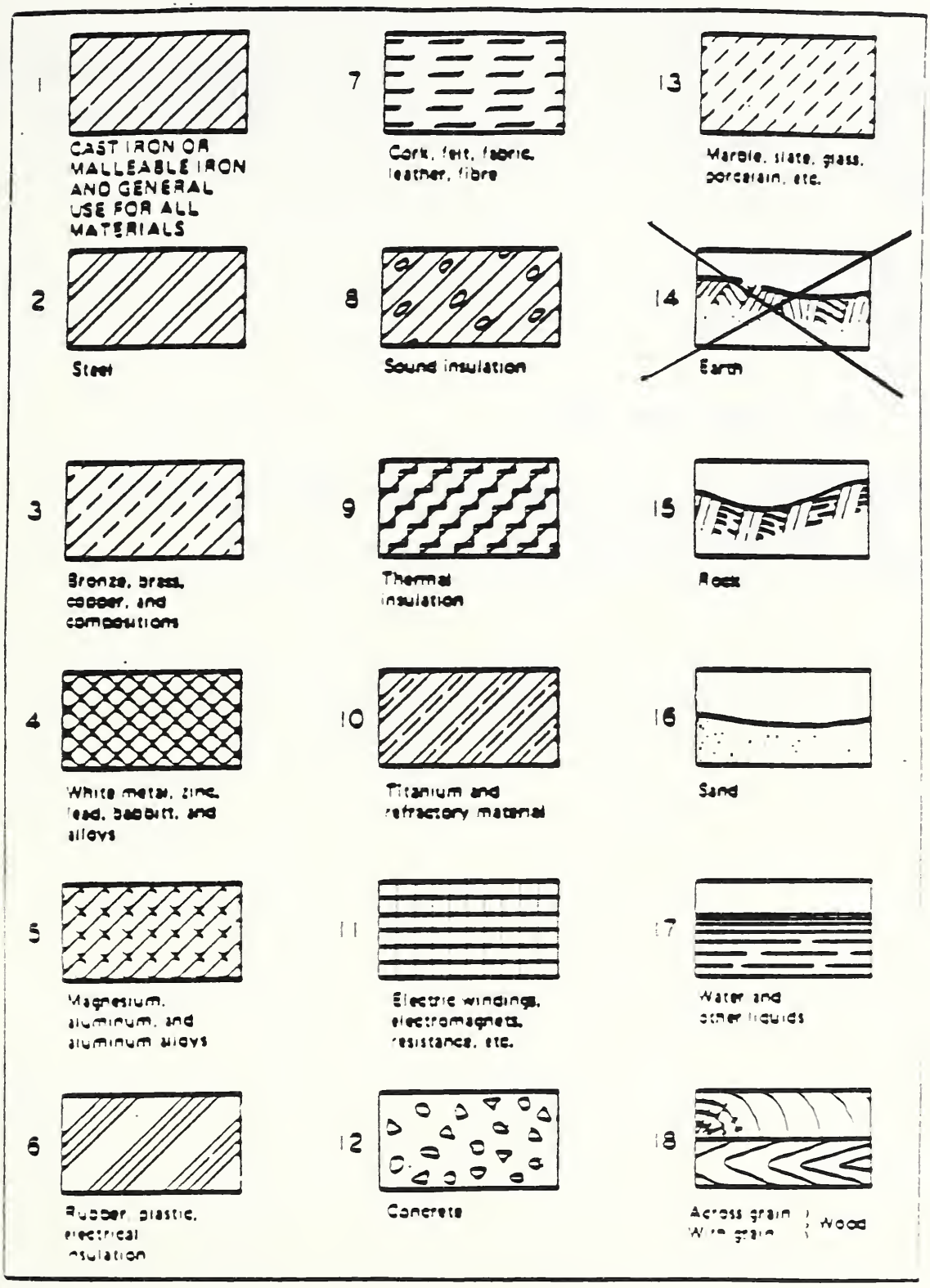


FIGURE 11: Examples of hatch styles

Custodians:
Army - CR
Navy - SH
Air Force - 24
DLA - DH

Review activities:
Army - AM
Air Force - 01,02
NSA - NS
DCA - DC
NASA - NA
Others - NBS, DOE, GPO, NCS

User activities:
Army - AL, AT, AV, EA, ER, GL, ME, MI, MR, SM, TE, TM
Navy - AS, EC, OS, SA, YD
Air Force - 11, 13, 14, 17, 18, 19, 68, 79, 99

Preparing Activity
OSD-CL
(Project ILSS - 0034)

GRAPHICS, CGM MIL-SPEC

Extended CGM MIL-SPEC Planning

CALS SOW TASK 3.1.2.4

FINAL REPORT

CALS SOW TASK 3.1.2.4

EXTENDED CGM MILSPEC PLANNING

TABLE OF CONTENTS

I. PURPOSE 1

II. BACKGROUND 2
1.0 Introduction 2
1.1 Motivation 2
1.2 Historical Review 2

III. DISCUSSION 5
1.0 Project Progress in 1988 5
2.0 Key Meetings of 1987-88 5
2.1 The Berlin SC24 Meeting 5
2.2 The Salt Lake Meeting 6
2.3 About Addendum 2 8
2.4 The Blakeney SWG/FP Meeting 8
2.5 The Blakeney Metafile RG Meeting 10
2.6 The Fairfax X3H3 Meeting 12
2.7 The Tucson SC24 Meeting 14

IV. SUMMARY AND CONCLUSION 16

V. RECOMMENDATIONS 18

GLOSSARY 19

APPENDICES
APPENDIX 1--ADDENDUM 3 INPUT FOR BERLIN SC24 MEETING . 21
APPENDIX 2--ADDENDUM 1 PDAD TEXT 39
APPENDIX 3--U.S. PDAD BALLOT RESPONSES 127
APPENDIX 4--SUMMARY ARTICLE OF THE BLAKENEY MEETING OF THE SPECIAL WORKING GROUP ON FUTURE PLANNING 155
APPENDIX 5--FIVE-YEAR PLAN PRODUCED BY SWG/FP. 159
APPENDIX 6--DOCUMENTS FROM BLAKENEY METAFILE RG MEETING 173
APPENDIX 7--ADDENDUM 3 DRAFT FROM FAIRFAX MEETING. . . 191
APPENDIX 8--METAFILE REFERENCE MODEL AND POSITIONS . . 249
APPENDIX 9--TUCSON METAFILE REPORT 261
APPENDIX 10--TUCSON RESOLUTIONS 271

EXTENDED CGM MILSPEC PLANNING

I. PURPOSE

Represent and inject CALS requirements in the standards efforts for the Extended CGM, or CGEM. Plan for development of a CGEM MILSPEC (CALS SOW Task 3.1.2.4).

This work was accomplished through a contractor, Mr. Lofton Henderson, of Henderson Software, who is a member of the following:

- the Accredited National Standards Committee X3H3 on Computer Graphics Standards;
- the sub-committee, X3H3.3, Computer Graphics Metafile, and Computer Graphics Interface (CGI);
- the U.S. delegation to the International Standards Organization Working Group 2 (ISO/WG2), Computer Graphics Standards;
- the ISO/WG2 CGM Rapporteur Group, responsible for processing comments, refining the standard, and issuing interpretations of the standard; and
- the ISO/WG2 sub-group developing the Extended Metafile standard.

As is apparent, Mr. Henderson is in a unique position to ensure that CALS requirements get addressed and injected into the extended metafile (CGEM) work at the national and international levels. He will be referred to in the remainder of this report as the NIST representative, which serves to properly identify the role that he has played in furthering, under NIST (the National Institute of Standards and Technology, formerly NBS) direction, the needs of CALS in the development of CGEM.

[NOTE: Please consult the GLOSSARY on pages 19 and 20 of this report for any acronyms that are unfamiliar.]

II. BACKGROUND

1.0 Introduction

This task was being pursued primarily at the working meetings of the graphics and metafile experts of ANSI and ISO. In addition, there was some significant work between meetings, preparing and pulling together position papers, baseline standards documents, and input documents to key meetings. The final report of the 1987 predecessor of this project mentioned briefly the ANSI meeting at Lowell, MA in October 1987. This report covers activities subsequent to the Lowell meeting.

The major emphasis of this task is on the CGM addenda. There has been some work on the related Graphical Registration proposals as well.

1.1 Motivation

One consequence of the consensus process for making the CGM standard is that the standard tended toward being a "least common denominator" metafile for the various constituents. It is, to a large degree, the area of overlap that all participants in its formulation agreed should be in a graphical metafile. As a result, it is relatively lean in functionality.

This expedited processing of the standard. However, CGM could be more efficient in some application environments. The CALS application areas of technical illustration, technical publications, and compound document exchange are such environments.

An extension process was immediately commenced to extend CGM functionality as required by more advanced metafile applications.

1.2 Historical Review

Two addenda were officially commenced in ISO:

Addendum 1: intended to support the GKSM requirements of GKS, and replace the non-standard specification in annex E of GKS;

Addendum 2: intended to support the metafile requirements of GKS-3D, and replace the non-standard annex E of GKS-3D.

Due in part to the efforts and success of the 1987 predecessor of this project, during 1987:

- The scope of Addendum 1 was expanded to support some of the advanced requirements of CALS (and similar constituencies): symbol libraries, additional geometric primitives, and basic raster primitives. The functionality was taken from and was compatible with CGI;

- "Addendum 3" was commenced, to extend CGM further as required by the technical illustration and publishing.

In early 1987 the NIST representative worked within ANSI to generate domestic support for such extensions. These positions were then taken input into the ISO meetings. The results were:

- o acceptance of broadening the scope of Addendum 1 as detailed above;

- o rejection (or postponement) of commencing work on Addendum 3.

While there was considerable support for the concepts of Addendum 3, there were procedural objections and some disagreement over what group should do the work and when.

This put the responsibility for the initiative on Addendum 3 back into the U.S. standards group, ANSI X3H3. The final activity of this project in 1987 was at the NIST/Eurographics workshop at Gaithersburg, MD, in September 1987. At this meeting a sub-group studied the requirements of CALS and similar technical constituencies. The following list summarized the requirements which were identified:

1. Internal symbol libraries;

2. Reference to and invocation of pre-defined external symbol libraries;

3. Advanced geometric drawing capabilities, including:
   - user defined linetype;
   - user defined hatch style;
   - a number of additional linetypes;
   - a number of additional hatch styles;
   - several types of spline curves;
   - conics and conic arcs;
   - closed figure primitive;
   - arbitrary clipping boundary;

4. a number and variety of fonts;

5. a completely new text model based on the work of ISO 9541;

6. additional raster primitives (and associated attributes for image processing);

Shortly thereafter the metafile sub-group of X3H3.3 met and produced the following for input to the initial SC24 meeting at Berlin in December 1987:

- A statement of need, scope and purpose for Addendum 3;
- A baseline document for the functionality of Addendum 3, in the style of Clause 5 of CGM Part 1 (abstract specification of the functions in Addendum 3).

In overview, in 1987 this project:

- organized ANSI support for the CGM extensions required for CALS;
- accomplished some of these extensions in the context of CGM Addendum 1;
- brought Addendum 3 for the first time into the ISO arena; and
- produced the groundwork (the requirements statement) for progressing Addendum 3 in ANSI and ISO.

III. DISCUSSION

1.0 Project Progress in 1988

The goals for this CALS CGEM task were:

1. Acceptance of the Addendum 3 project by ISO. Pursuing it as purely an ANSI project, and doing it on the fast track, was discussed. However there is significant enough interest in ISO that an ISO project would likely be initiated before such an ANSI project could complete. Past experience indicates that ANSI would not likely approve the content of Addendum 3 as an ANS if there were an active ISO project and if there were a chance of incompatibility between the two. Hence proceeding without ISO in this case risked the waste of significant effort, and ultimately the slowing down of the project.

2. Production of scope and purpose documents, and a reasonably complete baseline document for the Addendum 3 functionality. In the worst case, it was necessary to be prepared to proceed with standardization if ISO dragged on the project.

3. Circulation of Addendum 1 PDAD or Proposed Draft Addendum (see the Glossary) text in ANSI and generation of a U.S. position on the ISO PDAD ballot. There were two sub-goals here: insure that Addendum 1 was processed rapidly and smoothly; watch over the "CALS content" of Addendum 1 and insure that it remained intact.

4. Generation of some U.S. position on the 3D Addendum 2. There has been relatively little interest in this in the U.S. Still, it is in CALS interest to see the work go as smoothly as possible so that resources (which are scarce) are not drawn away from CALS priorities excessively.

2.0 Key Meetings of 1987-88

2.1 The Berlin SC24 Meeting

This was the first meeting of the SC24 Plenary. Between October and December a baseline document for the functional content of Addendum 3 was produced by the X3H3.3 metafile specialists. A draft of the functional content was produced as input to the Berlin meeting. Copies of the scope and purpose document and that of the baseline functional specification are in Appendix 1 of this report.

The goal at the Berlin meeting was to get Addendum 3 accepted as an ISO project and assigned to the WG3 Metafile Rapporteurs Group

(MRG). The input documents were tentative, but they were primarily intended to demonstrate serious interest, progress, and willingness to do the work. The Berlin meeting was not attended by any of the X3H3.3 CGM experts, but Peter Bono presented and represented the U.S. positions.

SC24 did not accept and initiate a project for Addendum 3. Instead two things of significance happened. It voted to circulate the documents for comment as a New Work Item (NWI) proposal. It set up a Special Working Group on Future Planning (SWG/FP) to study current SC24 projects and procedures, make recommendations, and recommend future projects.

There were several sources of opposition to initiating a fast project on the functional content of Addendum 3:

1. Some maintained that it was procedurally inappropriate to continue to initiate work as extensive as Addendum 3. Rather, the ISO NWI procedures should be followed;

2. There was sentiment, based on a certain reference model of computer graphics, that such significant functional extension cannot be standardized in the metafile without first standardizing it in the API (Application Programmer Interface) standards such as GKS. This reference model purports that standard metafiles can only be generated through standard interfaces, and cannot (for example) be used to transfer pictures in a standard manner to/from proprietary CAD systems.

3. There is reluctance to allow the MRG experts alone to design significant functional extensions (splines, conics, text, etc.) that may in the future be included in other standards. According to this point of view, such functionality really belongs to the "next generation" of graphics standards and should have wider participation in its formulation.

The results of the Berlin meeting were disappointing in that Addendum 3 was not accepted for progression. It was hoped that the Salt Lake ANSI meeting (January 1988) might be used as the first ISO working meeting on Addendum 3, with international members of the MRG attending. On the other hand, the results were encouraging in that the SWG/FP seemed to present an opportunity to get the work endorsed and initiated.

2.2 The Salt Lake Meeting

In December the PDAD text of Addendum 1 was received in the U.S. from the document editor (Anne Mumford) in England and the ISO PDAD ballot (3 months) was commenced. The document was circulated to X3H3. There was not time to have a letter ballot

before Salt Lake.

Because all open technical issues had been resolved, and because the PDAD document was supposed to reflect these resolutions, the position taken was that the U.S. response was a matter of verifying that the document did in fact reflect the resolutions. Consequently, because there were no unresolved technical issues, it was concluded that a working group of metafile experts could prepare the response at Salt Lake, and ask X3H3 plenary to approve it.

In fact the U.S. response came to over 25 pages and its production consumed the entire Salt Lake meeting. A copy of the Addendum 1 PDAD text is in Appendix 2 of this report, and the U.S. response document is in Appendix 3. There were no serious technical problems with the Addendum 1 PDAD text. The technical issues had been resolved in 1987 in a manner satisfactory to the U.S. (and CALS). Rather, there were extensive editorial problems and minor technical issues.

The most pervasive problems derived from the relationship of Addendum 1 to CGI (most of the technical material was lifted from CGI). These problems were twofold. Firstly, some CGI functionality changed at Valbonne and at an editing meeting a couple months later, but it was not possible to obtain new CGI text until mid-1988. The CGM Addendum 1 document editor had to make do with the old text. Secondly, much of the text (e.g., segmentation) was lifted almost verbatim from CGI. While the text was appropriate for a procedural interface, it was not appropriate for a graphical database definition such as CGM.

Finally, there were numerous problems that were oversights, omissions, etc. The only really significant new material that the Salt Lake review turned up had to do with the Formal Grammar. For the first time in several years there was a specialist on hand, and he detected a number of problems both with the Addendum text and the original CGM standard.

The response document was approved as planned by X3H3 plenary, and was sent to ANSI during February for forwarding to ISO. Although the U.S. had no serious objections to PDAD 1, it was voted "no". This is a procedural requirement of ANSI if any of the comments are technical, then the vote must be "no'. The "no" vote is reversed at the ISO meeting when the technical comment is resolved to the satisfaction of the U.S. delegation.

A final significant result at Salt Lake is that the NIST representative was selected as one of the two U.S. delegates for the SWG/FP meeting (see below).

2.3 About Addendum 2

There was an Addendum 2 (3D) working meeting in England in November, attended only by U.K. and Germany. Revised Addendum 2 text was supposed to be received by early February. The Berlin SC24 meeting had resolved that the text should be circulated as a Working Draft, for a comment period that expired in April. The Salt Lake X3H3 empowered an ad hoc group to meet and prepare a U.S. response. This meeting was planned to occur at the NCGA '88 expo in late March. In fact, the document was not received until the day before NCGA. There was no time to prepare a response, so the U.S. requested an extension of the comment period.

When the Addendum 2 WD text arrived, the minutes of the meeting in England had still not been received and there was little idea what to expect. The text that arrived was significantly broader in scope than what had been resolved. Instead of minimal support for GKS-3D, there was added support for a PHIGS MO workstation and even a replacement for the PHIGS archive file. The X3H3.3 metafile group, of which the NIST representative is a member, tended to not have strong positions on 3D, but the group knew there would probably be some strong feelings in X3H3.1 (the PHIGS sub-group). Therefore, the X3H3.3 group circulated the document to X3H3.3 and X3H3.1 and arranged for a joint ad-hoc group to prepare the U.S. response at the Fairfax X3H3 meeting (May 1988).

This radical change of scope of Addendum 2 turned out to be one of the triggering events for work at Fairfax and Tucson (SC24, July 1988) which changed the fundamental structure and direction of the addendum work.

2.4 The Blakeney SWG/FP Meeting

For some time there had been dissatisfaction within SC24 (and its predecessor SC21/WG2) over the state and process of making graphics standards. In particular:

- the process takes much too long;
- there is insufficient coordination among projects that are supposed to be closely related;
- there is lack of an overall reference model explaining the relationships between the standards.

The list goes on with a number of other points. Coupled with this dissatisfaction, there was sentiment within SC24 to shelve CGI (it was taking too long and had missed its "window of opportunity") and demands for a number of new standards and extensions even before work on the "first generation" of standards was complete. These new projects included:

1. CGM Addendum 3;
2. PHIGS extensions (PHIGS+);
3. Windowing standards;
4. Imaging standards;
5. GKS review and revision;

In order to sort out these competing demands the Berlin SC24 meeting established a Special Working Group on Future Planning. Each national body was allowed to send two representatives. The NIST representative was selected to be one of the U.S. representatives. This looked to be an excellent strategic outcome for Addendum 3 constituents, which includes the CALS constituency, because it was expected that the SWG/FP would deal specifically with new proposals such as Addendum 3. Our NIST representative was thus in a good position to influence the progress and acceptance of Addendum 3 as an ISO project.
In fact, during the 3-day meeting (11-13 April) it was never possible to pull the attention of the SWG/FP down to the level of talking about specific projects. Instead there was considerable discussion of:

- "lessons learned";
- a recognition of the distinction between first and second generation standards;
- a determination that first generation standards must finish quickly (within a year); and
- a determination that all new work must be structured differently.

The IEEE Computer Graphics and Applications magazine article by G.S. Carson provides an excellent summary of this meeting. Rather than repeat that material, the article is included in Appendix 4. In Appendix 5 is a copy of the 5-year plan that is referenced in the article.

CGM Addenda 1 & 2 were considered to be first generation standards, and were encouraged to complete work rapidly. At the same time it was recognized that Addendum 2 was in some trouble, as there was widespread dissatisfaction over the change of scope and apparent confusion of purpose--there was skepticism that it could reach DIS stage in the 1-year period before the cutoff, given that it had not even managed to reach an agreed scope yet. The U.S. (and CALS) would not have minded work being suspended on Addendum 2, since it is perceived as draining scarce resources that could be used on Addendum 3.

Addendum 3 was officially considered to be a second generation standard. The NIST representative observed a reluctance to deal

with it. This was for the reasons discussed above (the Berlin SC24 meeting). There was also some sentiment from GKS advocates that the work belonged in GKS extension and review. There was the already mentioned philosophical objection that functionality could not be included in a metafile standard that did not have correspondence in an API (Application Programmer Interface) standard.

In term of officially results, then, SWG/FP did little for advancing Addendum 3. Unofficially, however, there was some productive exchange with other national delegates. The NIST representative was able to communicate two points in particular:

1. The real-world experience gained in the multi-vendor CGM system integration demonstration at NCGA's Integrate '88 was used as evidence for the U.S. position that a metafile standard is not only used just by **standard** application systems, and can have functionality not seen in such systems. At Integrate '88 it is estimated that not more than 1/3 of the CGM generating and interpreting systems involved other ISO graphics standards. Rather, CGM modules were attached to existing proprietary systems, such as CAD systems. Some key objectors to Addendum 3 privately admitted that this view of metafile utilization had merit.

2. It was not concealed that the U.S. would proceed on the standardization of Addendum 3 if ISO did not demonstrate interest in expediting such work. It was the U.S. position that the normal 5-year timeframe is much too long to wait for this work. The extensions are needed now.

By the time of the Tucson meeting there appeared to be some acknowledgement of these points. The NIST representative thought that the unofficial exchanges at Blakeney had some effects. The work of the SWG/FP and subsequent national body contributions formed the framework and defined the agenda for that meeting.

2.5 The Blakeney Metafile RG Meeting

In the week following the SWG/FP meeting there was a 3-day (18-20 April) WG3 Metafile Rapporteur Group (MRG) meeting at the same location in Blakeney, England. The purpose of the meeting was twofold: firstly, the PDAD ballot on Addendum 1 was complete and those results had to be processed; secondly, it was planned to process WD comments on Addendum 2 and prepare to advance the text to PDAD stage.

The meeting was not officially empowered as an editing meeting, so could not produce official response document. Rather the output would be input to the official MRG editing meeting at Tucson. In reality, since the active metafile experts from the

most important delegations were present, it was anticipated that the work would be endorsed mostly intact at Tucson. The meeting was sparsely attended: 1 U.S. (NIST representative), 3 U.K. (but only 2 present at any given time), and 2 Germany.

The time allocated for the two addenda was 1 day for PDAD Add.1 comments, and 2 days for refining Add.2. We (the U.S.) objected that 2 days was too much for Addendum 2 the document had only been circulated 3 weeks before and none of the national comments were in hand. It was also pointed out that it was known that there were objections from several nations over the expansion and confusion of scope. It was agreed to spend more time on Addendum 1, spend an afternoon explaining and discussing Addendum 2, and spend the last day finishing off any issues on Addendum 1 and drafting output documents on both addenda.

The PDAD ballot drew 5 negative votes, including France, Germany, U.K., U.S. (see "Salt Lake Meeting"), and Czechoslovakia. Most of the objections were similar to those of the U.S., including especially concern over the reliance on CGI, which was perceived to be unstable. The MRG response on that point was basically: in consequence of the SWG/FP work, CGI must either be technically stable now or it won't become a standard. If Add.1 is simply made current with CGI, then CGM should be stable in those areas of overlap.

As with the U.S. response, other national responses contained many minor technical points. In fact it took all of the allocated time plus some to sort through the points and come to consensus on the important ones. Some were stacked for Tucson. From the CALS point of view, nothing significant happened at this meeting (which was just what was wanted--stability of the extended functionality which was important to CALS). There was some move to remove PIXEL ARRAY (an element needed by CALS) because of alleged changes in CGI. This discussion was postponed for Tucson. Aside from important functionality remaining intact, the proposed Addendum 1 schedule was the only other topic of importance to CALS: the MRG agreed to attempt an aggressively fast schedule--produce DIS text at Tucson and try for IS text in summer 1989.

There was no significant time available or allocated for Addendum 3. The best that could be achieved at this meeting was a weakly worded recommendation from the MRG to SC24 that the work is important and should be progressed (possibly by experts from several areas in addition to metafiles). This did serve to keep the topic alive and visible. But overall, the two Blakeney meetings did not produce any definitive new results for Addendum 3.

Appendix 6 contains selected documents from the Blakeney MRG meeting. Included are the minutes, the explanation of Addendum

2, and a summary of voting on PDAD Addendum 1.

2.6 The Fairfax X3H3 Meeting

Attempts to persuade ISO to immediately commence work on Addendum 3, and assign it to the WG3 metafile rapporteur group, had not achieved positive results at the Berlin and Blakeney meetings. It was the U.S. position that the U.S. would proceed if a project were not taken up by ISO. The purpose of proceeding within the U.S. was twofold: firstly, the constituency for Addendum 3 (e.g., CALS) needed the functional extensions without undue delay; secondly, ISO did apparently intend to take up the work sooner or later, in some form. ISO would likely take it up sooner if the U.S. continued to progress Addendum 3 as a domestic U.S. standard, and the subsequent technical work would proceed faster as well.

At the Fairfax meeting, then, the metafile sub-group had two goals with regard to Addendum 3. The first was to produce a new revision of the baseline document for Addendum 3. The second was to produce a draft SD-3 for an X3H3 Addendum 3 project. An SD-3 is the statement of scope, purpose, and need which is used to initiate a standards project in one of the accredited ANSI committees. The SD-3 was to be circulated during the summer for an approval ballot within X3H3. Approval would basically give the project the standing of an official ANSI project (after some higher approvals in X3).

The two documents were also intended for circulation within ISO. Again the purpose was twofold: to demonstrate that the U.S. was serious about the need for the project, was ready to supply the necessary resources, and was willing and capable of proceeding alone; and to keep ISO metafile experts current so that a transition could be effected smoothly should ISO commit to doing Addendum 3.

Some of the work to refine and extend the draft document was done by interim assignments before Fairfax. The rest was done at Fairfax and in working assignments during the following month. The result is contained in Appendix 7. A draft of the SD-3 was produced at the meeting. The metafile sub-group would have liked to be able to refine the drafts somewhat more, and to get more group discussion and issues resolution. Lack of time and staff was a problem.

There were 10 people available to work at Fairfax. It turned out that the NIST representative and one other had to work most of the three days on other matters.

One of the items on the agenda at the meeting was to form a U.S. position on Addendum 2, 3D. There was significant unhappiness

with the draft which had been received from England in March. It had significantly expanded in scope, to the point where it concerned U.S. 3D experts (PHIGS). More importantly, a number of U.S. experts within X3H3 were concerned with attaching the functionality of Addendum 2 and the GKS audit trail portion of Addendum 1 to the CGM standard. Regardless of the merits of the material, its nature is contrary to the basic philosophy of the static picture capture CGM. Within the U.S. there was a significant constituency which appreciated what the CGM was and found it useful and appropriate.

A liaison meeting was scheduled to work with the 3D experts to formulate a U.S. position on Addendum 2. In fact, the output of the meeting was the production of a basic Metafile Reference Model (MRM) which demonstrated why the U.S. was dissatisfied with what was happening in the addendum process. The MRM shows:

1. there can be metafiles defined at any level in a graphical hierarchy; and

2. the metafiles can either be static (snapshot) in nature, or they can be dynamic (session capture). The former are a snapshot of the state of some layer in the hierarchy, while the latter are an audit trail of the dialogue over some interface in the model.

According to the MRM the CGM is a static picture capture metafile at the level of the virtual device. The GKSM content of Addendum 1, however, is at a higher level in the model and is a session capture specification. Similarly, several distinct metafiles were identified in the draft Addendum 2--all were of a different nature (dynamic) or at a different level than CGM in the MRM.

This model was refined and approved by X3H3 for input to the Tucson meeting as a U.S. position (see Appendix 8). The model became the basis for U.S. objections to Addendum 2, and was the basis for a proposal that Addendum 1 be split. The static portion of Addendum 1 (the portion of interest to CALS) would continue to be progressed as an addendum to CGM. The audit trail portion would progress as an addendum to GKS. According to the MRM, Addendum 3 was in the spirit of CGM and should properly be progressed as an addendum to CGM. This is contrary to several opinions which were being heard in the ISO community.

Although an unexpected result, the MRM and the position papers on the addenda which derived from the MRM were by far the most important output of the Fairfax meeting and had the most far-reaching implications.

2.7 The Tucson SC24 Meeting

The MRM and associated position papers were pre-circulated to key members of the metafile rapporteur group prior to Tucson. Some of the other national delegations were having similar problems with the CGM addenda, and so the time was ripe for a proposal that made some sense of the disorder. There was interest in the U.S. proposals.

At the Tucson meeting itself, the MRM and the U.S. proposals for restructuring the CGM work were endorsed in the first half of the first day. The rest of the meeting was basically concerned with implementing the new structure, working out technical issues, and working out procedural problems and schedules. A detailed report of the activities at Tucson is contained in Appendix 9.

To summarize the practical effects of the Tucson resolutions (see Appendix 10 for the complete resolutions text):

- the static picture content of Addendum 1 continues to be progressed as an addendum to CGM. It is known as the static structured picture capture (SSPC) metafile. It contains everything from addendum 1 that is important to CALS. It is known as CGM Add.1.

- the dynamic session capture content of Addendum 1 is removed and progressed separately as an addendum to GKS, known as GKS Add.1. The timetable is to be the same as that for CGM Add.1. There is nothing in GKS Add.1 that is of much importance to the CALS constituency.

- the static content of Addendum 2 will be progressed as an addendum to CGM, known as CGM Add.2. It will represent static 3D pictures. There are some interesting possibilities for CALS in this content. In particular, it may be very natural to translate IGES and PDES into CGM Add.2 (or Add.2 extended with advanced geometric primitives).

- the dynamic content of Addendum 2 will be removed and progressed as an addendum to GKS-3D, known as GKS Add.2.

About Addendum 3: WG1 (reference models, user requirements, etc) and the Tucson plenary passed resolutions which finally gave standing to Addendum 3 in ISO. It is recognized that two technical areas are critical to Addendum 3: advanced text and font facilities; and advanced geometric primitives (splines, conics, etc.). It was also recognized that these technical areas are of key importance in other SC24 standards and standards proposals, such as GKS 9x (the successor to GKS), PHIGS+, etc.

Accordingly a number of technology and study groups were formed. Three of these of importance to CALS are:

1. advanced text models;

2. product data geometry;

3. CGM extensions.

The third group is essentially an Addendum 3 study group. It is to meet in parallel with the technology groups, decide whether a New Work Item is desirable, and produce such and a high-quality baseline document (draft Addendum 3 standard). The timeframe for completion of the study and baseline document is about one year. The rapporteur of the metafile extensions study group is the NIST representative.

This at last is the positive commitment to Addendum 3 that was sought from ISO. There may be some negative aspects to the ISO commitment. Foremost among these is: how long will it take to get the work done in ISO?

A final consequence of the Tucson resolutions was that there was no longer any particular rush to produce the SD-3 for Addendum 3. However, since final text for CGM Add.1 was scheduled for summer 1989, there was now some urgency to initiate formal ANSI processing. This had not been done previously, because ANSI is in the process of adopting new procedures which would allow for automatic tracking and adoption of ISO standards work (without having to carry out parallel ANSI work). These procedures have been very much slower in coming than was anticipated. Accordingly, the SD-3 for Addendum 3 was converted to an SD-3 for CGM Add.1, and this is being voted now in letter ballot by X3H3.

IV. SUMMARY AND CONCLUSION

The 1987 predecessor of this task was largely concerned with defining the CALS requirements for CGEM, getting those requirements endorsed and agreed to by both the ANSI and ISO bodies working on the extensions, and getting work initiated on the CALS extensions. The requirements definition and ANSI X3H3 endorsement were largely accomplished. This task has continued to seek ISO endorsement and status for the project and has otherwise focussed on producing draft documents containing the required functional content.

Specific goals for 1988 were:

1. Expedite processing of CGM Addendum 1. As a result of work in 1987 Addendum 1 contained a number of functions that were important and useful to CALS.

2. Get ISO approval for Addendum 3 as an ISO project, assigned to the existing Metafile Rapporteur Group (MRG), and "fast-tracked" to completion. Most of the required CALS functionality resides in Addendum 3.

3. Coordinate the CGEM work with registration proposals to insure maximum compatibility. While awaiting the more permanent results from the lengthier CGEM standardization process, CALS is also pursuing a short term strategy of submitting some of its required functionality for Graphical Registration.

For CGM Addendum 1, progress was excellent in 1988. Addendum 1 was submitted for its first PDAD ballot in 1988. The U.S. submitted a 25-page set of comments, participated in the preliminary processing of the ballot results at an interim ISO Metafile Rapporteur Group meeting, and participated in the final processing at the June ISO SC24 meeting. Between these two meetings an ANSI X3H3 working meeting produced a Metafile Reference Model, which was submitted to ISO. This led to a complete restructuring of the work on Addendum 1 (and Addendum 2 as well). The static extended functionality and static segments important to CALS will continue to be processed as an addendum to CGM, while dynamic functionality (of little use to CALS) will be split out and progressed as an addendum to GKS. Addendum 1 will progress directly to DAD status, bypassing an anticipated second PDAD ballot. Assuming no unforseen technical problems on the DAD ballot, final text should be available in about 1 year.

No new CALS functionality was added to Addendum 1, but this was neither intended nor desired. Basically the functional content of this addendum was closed in late 1987, CALS requirements were included as much as feasible, and the 1988-89 CALS agenda for Addendum 1 is to get the processing completed as soon as

possible.

Most of the CGM extensions needed by CALS are addressed in Addendum 3. On Addendum 3 results were mixed. ANSI X3H3 has endorsed the project and indicated willingness to pursue it as a U.S. standard if ISO does not pick it up. By mid-year a draft of the required functional content had been produced, complete with Scope and Purpose, Functional Specification, and the three data encodings. The SD-3 had also been prepared, which is the document required to give the project official status as an ANSI standards project. A projected timetable showed completion of the project in 1990.

There has been resistance within ISO to initiating the project as an addendum to CGM, and this continued through mid-year. No official endorsement of the project was achieved in the first few opportunities. In fact there was positive resistance, based primarily on political and "jurisdiction" issues. Finally at the SC24 meeting in June a study group was established, which should lead to a New Work Item and DP-quality draft within a year. This is both good and bad news. On the positive side, ISO has now given status to the project, and the study group rapporteur is the NIST representative. Thus CALS is in an excellent positions to see that its requirements are included in the new work. On the negative side, the process will probably be slower by 1-1.5 years than if ANSI had proceeded alone. It may be 3 years until final text is available. In the balance, ANSI could probably not have proceeded alone without getting overtaken by ISO at a later date.

V. RECOMMENDATIONS

The CALS required content of Addendum 1 is fairly safe now, and routine oversight of remaining Addendum 1 processing should suffice (the 6-month DAD ballot commences in Oct-Nov 1988). The requirements study for "Addendum 3" is complete and draft text has been circulated within X3H3 for the functionality. However the standard embodying this will be an ISO standard, and its scope and content will be completely determined during the next 12 months. During this period it is critical for CALS to continue to present its requirements. An ideal opportunity to do so exists, as the NIST representative has been approved to lead the study group for these extensions.

Recommendation: NIST must take advantage of the present opportunity to inject CALS requirements during the study and definition phase for the new ISO CGM extensions work. The NIST representative must be funded to participate in domestic and international CGM meetings on CALS behalf.

Recommendation: Due to the longer than anticipated timeframe of the desired CGM extensions, NIST suggests that the scope of the NIST representative's task in FY 89 include a reassessment of extending the CALS AP (MIL-D-CGM) in the interim by inclusion of registered graphical items as these become stable. This recommendation implies that planning a MILSPEC for CGEM (that is, Addenda 1, 2 and 3 of CGM) is premature at the present time. Rather, more time is necessary to evaluate how long the standards process is likely to take before committing to a MILSPEC for CGEM. The plan that was asked for now should be a product of the coming fiscal year, since it depends so much on the stability and timeliness of these Addenda within the standards process. Either of the following scenarios is a possibility:

(1) Continue adding functionality to MIL-D-CGM via stable registration proposals and/or Addendum 1. (This is what the above recommendation is for FY 89.)
(2) Use CGEM Addendum 1, along with the more stable registration proposals, as the basis of a first draft MILSPEC CGEM, and then build in the other Addenda as they become standardized.
(3) Continue as in step 1, but instead of having 2 MILSPEC's, have one in future (in probably 2 to 3 years time) which includes MIL-D-CGM as well as the standardized versions of CGEM Addenda, thereby superseding MIL-D-CGM (if allowable in DOD).

GLOSSARY

ASC X3H3 Accredited Standards Committee X3H3, the ANSI accredited committee responsible for computer graphics standards in the US.

X3H3.3 The subcommittee of X3H3 that is responsible for CGM and CGI.

ISO/IEC JTC1/SC24 International Standards Organization, Joint Technical Committee 1, Standing Committee 24, the international counterpart to X3H3.

ISO TC97/SC21/WG2 The predecessor to SC24 (prior to December 1987).

WG3 The working group of SC24 responsible for standards work in metafiles and device-level interfaces, i.e., CGM and CGI.

Metafile Rapporteur Group (MRG) The sub-group of WG3 responsible for CGM maintenance and CGM extensions.

Working Draft (WD) The first complete draft of a proposed ISO standard, the starting document for subsequent work and review.

Draft Proposal(DP) The second stage in the ISO processing pipeline. After national bodies have commented on the WD, it is altered and refined and then registered as a DP. Another round of ballot and comment takes place on the DP.

PDAD Proposed Draft Addendum, the same as DP, but for an addendum as opposed to a standalone project.

Draft International Standard (DIS) The project stage in the ISO pipeline after DP. The technical content of the project is supposedly highly stable and it is expected that IS text can be produced subsequent to processing the DIS ballot results.

DAD Draft Addendum, the same as DIS, but for an addendum as opposed to a standalone project.

International Standard (IS) The final stage in the ISO pipeline, nothing remains but possibly the printing.

American National Standard (ANS) The final stage in the ANSI

pipeline, nothing remains but possibly the printing.

CGM — Computer Graphics Metafile, ANSI standard X3.122-1986 and ISO standard ISO 8632/1-4 1987.

CGI — Computer Graphics Interface, another ANSI/ISO standards project, currently at the 2nd DP stage. CGI is an interface standard which exists about at the level of the CGM in the graphics pipeline (device level). CGI is an interactive (input) and highly extended and enriched interface specification, whereas CGM has output-only functionality (for picture definition) and is a picture description protocol (a graphical database). CGI embeds CGM output functionality as a subset.

GKS — Graphical Kernel System, an application programmer interface to computer graphics, now an ANSI and ISO standard.

GKSM — Metafile for use with GKS. One was proposed in non-standard Annex E of GKS. Work on it was deferred in favor of CGM, and now of extended CGM (CGEM).

CGEM — Computer Graphics Extended Metafile, a set of addenda and extensions to CGM, being processed by ISO, currently nearing DP stage.

BSI — British Standards Institute, the British equivalent of ASC X3H3.

DIN — The German equivalent of ASC X3H3.

AFNOR — The French equivalent of ASC X3H3.

APPENDIX 1

ADDENDUM 3 INPUT FOR BERLIN SC24 MEETING

Purpose

The purpose of this addendum is to extend the CGM to effectively fulfill the picture transfer requirements of:

1) Engineering drawing and technical illustration
2) Graphics arts quality pictures, including geometric graphics, raster images, and text
3) Technical publishing

An additional intent is to keep pace with the graphics requirements of office systems, especially ODA requirements.

Scope

This addendum comprises a set of elements which will extend the capabilities of CGM as needed to meet additional user requirements in engineering drawing, graphics arts, and technical publishing. The set of elements should include all elements necessary to meet those requirements. It should be the minimal set sufficient to meet those requirements effectively.

The following preliminary list of capabilities is identified as necessary to meet these requirements.

1) Advanced 2D graphics, to include:
   - Bezier curves
   - Rational B-splines
   - Parametric spline curves
   - Line attributes of cap, miter, and join
   - Composite line primitive
   - User-defined line types
   - User-defined hatch styles
   - Arbitrary text path
   - Conics, and conic arcs

2) Text and font model of ISO 9541, Information Processing--Font and Character Information Interchange.

3) Picture composition and control, to include:
   - Arbitrary clipping boundary (general closed curve)
   - Shielding
   - Alignment

4) Additional color models beyond RGB
   - CIE
   - CMYB
   - Named colors

5) Additional raster graphics (scanned image) capabilities

6) Symbols: external reference to "standard" libraries of named symbols

The scope of this addendum assumes that the capabilities of CGM Addendum 1 and Addendum 2 are available.

The remainder of this document constitutes the preliminary work which has been performed to date.

END COMPOUND TEXT PATH delimits the end of a compound text path definition.

References:

5.X.X BEGIN CLIP REGION

Parameters:

none

Description:

BEGIN CLIP REGION delimits the beginning of a definition of a entity that will provide the clipping region. When CLIP INDICATOR is 'on' only the portions of graphics elements inside or on the boundary of the clipping region are drawn. The elements that make up the clipping region can be any combination of closed or non-closed elements such as POLYLINE, DISJOINT POLYLINE, POLYGON, POLYGON SET, CIRCULAR ARC 3 POINT, CIRCULAR ARC 3 POINT CLOSE, CIRCULAR ARC CENTRE, CIRCULAR ARC CENTRE CLOSE, ELLIPTICAL ARC CLOSE, or <new curve elements>. The entity thus defined is essentially a closed figure whose boundary is used as the clipping boundary.

Once defined, the clipping region takes the place of the clipping region defined in CLIP RECTANGLE.

References:

5.X.X END CLIP REGION

Parameters:

none

Description:

END CLIP REGION delimits the end of a clipping region definition.

References:

5.X.X BEGIN SHIELD REGION

Parameters:

none

Description:

BEGIN SHIELD REGION delimits the beginning of a definition of a entity that will provide the shielding region. When SHIELD INDICATOR is 'on' only the portions of graphics elements outside of the shielding region are drawn. The elements that make up the shielding region can be any combination of closed or non-closed elements such as POLYLINE, DISJOINT POLYLINE, POLYGON, POLYGON SET, CIRCULAR ARC 3 POINT, CIRCULAR ARC 3 POINT CLOSE, CIRCULAR ARC CENTRE, CIRCULAR ARC CENTRE CLOSE, ELLIPTICAL ARC CLOSE, or <new curve elements>. The entity thus defined is essentially a closed figure whose boundary is used as the shielding boundary.

References:

5.X Control Elements

5.X.X BEGIN COMPOUND LINE

Parameters:

none

Description:

BEGIN COMPOUND LINE delimits the beginning of a definition of a entity that will have consistent line attributes and will be treated as a single "compound primitive". The elements that make up the compound line can be any combination of non-closed line elements such as POLYLINE, DISJOINT POLYLINE, CIRCULAR ARC 3 POINT, CIRCULAR ARC CENTRE, ELLIPTICAL ARC, or <new curve elements>.

References:

5.X.X END COMPOUND LINE

Parameters:

none

Description:

END COMPOUND LINE delimits the end of a compound line definition.

References:

5.X.X BEGIN COMPOUND TEXT PATH

Parameters:

none

Description:

BEGIN COMPOUND TEXT PATH delimits the beginning of a definition of a entity that will provide the path in which a text string will be drawn. The elements that make up the compound text path can be any combination of non-closed line elements such as POLYLINE, DISJOINT POLYLINE, CIRCULAR ARC 3 POINT, CIRCULAR ARC CENTRE, ELLIPTICAL ARC, or <new curve elements>.

Once defined, the compound text path takes the place of the text path as defined by the TEXT PATH element and the CHARACTER ORIENTATION elements. The skew of the characters is still relative to that specified in the CHARACTER ORIENTATION element, but the placement of subsequent characters is along the compound text path instead of in a line along the character up vector or character base vector.

References:

5.X.X END COMPOUND TEXT PATH

Parameters:

none

Description:

5.X.X END SHIELD REGION

Parameters:

none

Description:

END SHIELD REGION delimits the end of a shielding region definition.

References:

5.X.X SHIELDING INDICATOR

Parameters:

shield indicator (one of: off, on) (E)

Description:

When SHIELD INDICATOR is 'off', shielding of graphical primitive elements is not required.

When SHIELD INDICATOR is 'on', only those portions of graphical primitive elements outside of the shielding region are drawn.

References:

All of the fontmetric elements (FONTMETRIC DEFINITION LIST, CHARACTER KERNING MODE, CHARACTER KERNING TABLE, FCS TYPE, FCS INTEGER PRECISION, FCS REAL PRECISION, and FCS EXTENT) are Metafile Descriptor Elements.

5.X.X FONTMETRIC DEFINITION LIST

Parameters:

list of:

font index (IX)
character index (C)
left bearing (format to be determined)
right bearing (format to be determined)
character height (format to be determined)
offset from baseline (format to be determined)

Description:

The fontmetric information for each character used in each font specified is defined by this element. If this element is used, then the fontmetric data for each character used in the metafile must be specified. Characters not used by the metafile may also be specified, but are not required.

References:

5.X.X CHARACTER KERNING MODE

Parameters:

character kerning mode (one of: none, pair, sectored) (E)

Description:

Defines the kerning style, if any, for the metafile.

References:

5.X.X CHARACTER KERNING TABLE

Parameters:

To be determined.

Description:

The data defined by this element will be dependant upon which, if any, kerning styles are supported. In general, however, the information will be that which is required to kern characters. The most prevalent form of kerning is character pair kerning.

References:

5.X.X FCS TYPE

Parameters:

Font coordinate type (one of: integer, real, VDC) (E)

Description:

The single parameter is an enumerated value that declares the data type, integer or real, of the font coordinate space. Font coordinate

space may be different than VDC space because higher precision may be necessary to accurately define the fontmetric data. However, in the font coordinate type is VDC, then the font coordinate space will map to VDC space. In this case, no additional specification for font coordinates (FCS INTEGER PRECISION, FCS REAL PRECISION, or FCS EXTENT) need be specified.

References:

5.X.X FCS INTEGER PRECISION

Parameters:

Encoding dependant.

Description:
The indicated integer precision for fontmetric data. The precision is defined as the field width measured in units applicable to the specific encoding.

References:

5.X.X FCS REAL PRECISION

Parameters:

Encoding dependant.

Description:
The indicated real precision for fontmetric data. The precision is defined as the field width measured in units applicable to the specific encoding.

References:

5.X.X FCS EXTENT

Parameters:

first corner (P)
second corner (P)

Description:
The two corners define a rectangular extent in font coordinate space that demarcates a "font window". Each character within a font must be contained completely within this window.

References:

5.X Attribute Elements

5.X.X Line Type Definition

Parameters:

linetype (IX)
dash unit selector (one of: VDC, mm, native device units, abstract) (E)
dash repeat length (R)
adaptive flag (one of: no, yes) (E)
list of dash elements (nI)

Description:
This element defines a linetype and associates it with an index for future reference. Parameter 'linetype' is the index of linetype being defined. The parameter 'list of dash elements' is the definition to be associated with the index. The first element is a dash, second a space, etc. -- the defined linetype is solid for I1 units, gap for I2 units, solid for I3 units, etc. N must be positive, and each dash element (I) non-negative. N=1 means a solid line; I=0 interpreted as a dot.

The units of the 'dash repeat length' are specified by the 'dash unit selector' parameter. The value of 'abstract' indicates that the implementation may normalize and map the sum of the dash pattern elements at its discretion. The 'dash repeat length' defines the length of one complete cycle of the dash pattern, measured in the units of 'dash unit selector'.

An "adaptive" linetype is one where every vertex falls on an inked portion of the line. This is accomplished in plotters by temporarily modifying the duty cycle for each line segment (ceiling function) such that there is always an integral number of repeats (and all predefined linetypes have their gaps_array defined such that they begin and end with inked or "pen down" portions).

References:

5.X.X Hatch Style Definition

Parameters:

hatch index (IX)
style indicator (one of: parallel, crosshatch) (E)
hatch space units selector (one of: VDC, mm, device units, abstract) (E)
angle (2R)
duty cycle length (R)
list of hatch elements (nI) - I>=0, n>=2

Description:
This element defines a hatch style and associates it with an index for future reference.

The 'hatch index' parameter defines the index of hatch style being defined. The 'list of hatch elements' is an array that defines alternating line width and gap width -- i.e., the width of a hatch line followed by the width of the space to the next hatch line. The center of the first hatch line is matched up with PATTERN REFERENCE POINT, if implemented. 0 interpreted as thinnest line width available.

The 'hatch space units selector' specifies the units of 'duty cycle length'. It also controls the manner of transformation of the hatching: If VDC, then the hatching transforms with segment transform and anisotropic transforms (as if hatching had done POLYLINES); otherwise, the hatching is like "wallpaper" that shows through the polygon-shaped hole -- everything is defined in device units and are doing hatching in device space. The value of 'abstract' indicates that the implementation may normalize and map the sum of the dash pattern elements at its discretion. The 'duty cycle length' is measured perpendicular to the hatch line. The sum of hatch elements in the hatch element list is normalized to this distance before presentation of the hatch on the view surface.

The 'angle' parameter is measured in the units specified by the 'hatch space units selector'. It consists of two components, dx and dy, defining a vector.

5.X.X Line Cap

Parameters:

line cap indicator (one of: butt, round, projecting square) (E)

Description:

The line cap style is defined for subsequent line elements. The line cap style determines the appearance of open endpoints (as opposed to interior vertices) of line elements. The defined styles are:

butt cap: the line is squared off at the endpoint, there is no projection beyond the endpoint.

round cap: a semicircular arc with diameter equal to the line width is drawn around the endpoint and filled in. The drawn line thus projects beyond the endpoint.

projecting square cap: the line is squared off at a distance equal to half the line width beyond the endpoint.

References:

5.X.X Line Join

Parameters:

line join indicator (one of: miter, round, bevel) (E)

Description:

The line join style is defined for subsequent line elements. The line join style defines the appearance of interior vertices of polyline elements and of compound line elements. The defined styles are:

miter join: the outer edges of the two adjoining line segments are extended until they meet at a point.

round join: a circular arc with diameter equal to the line width is drawn around the vertex between the adjoining segments and is filled in, producing a rounded corner.

bevel join: the adjoining line segments are terminated with a butt cap, and the resulting triangular notch is filled in.

References:

5.X.X Edge Cap

Parameters:

edge cap indicator (one of: butt, round, projecting square) (E)

Description:
The edge cap style is defined for subsequent edge elements. The edge cap style determines the appearance of open endpoints of filled area edges (such as may result from a mixture of visible and invisible edge segments). The defined styles are:

butt cap: the edge is squared off at the vertex, there is no projection beyond the endpoint.

round cap: a semicircular arc with diameter equal to the edge width is drawn around the endpoint and filled in. The drawn edge thus projects beyond the endpoint.

projecting square cap: the edge is squared off at a distance equal to half the edge width beyond the endpoint.

References:

5.X.X Edge Join

Parameters:

edge join indicator (one of: miter, round, bevel) (E)

Description:
The edge join style is defined for subsequent filled elements. The edge join style defines the appearance of interior vertices of filled area elements. The defined styles are:

miter join: the outer edges of the two adjoining edge segments are extended until they meet at a point.

round join: a circular arc with diameter equal to the edge width is drawn around the vertex between the adjoining segments and is filled in, producing a rounded corner.

bevel join: the adjoining edge segments are terminated with a butt cap, and the resulting triangular notch is filled in.

References:

5.X.X Miter Limit

Parameters:

miter limit (R)

Description:
Mitered corners can extend very far beyond the line vertex if the angle between the adjoining line segments is small. Miter length is defined to

be the distance from the point at which the inner edges of the adjoining line segments meet to the point at which the outer edges meet. If miter length exceeds the 'miter limit' parameter, then the joining line segments are rendered with a bevel join instead of a miter join.

Miter limit is measured as a scale factor applied to the current line width. Miter limit applies to line elements and edges of filled areas.

References:

5.X.X External Symbol

Parameters:

To be determined.

Description:

Reference to external defined symbol libraries is provided. The mechanism of this element is yet to be defined.

The following "Abstract Specification of Elements" is of the form in ANSI X3.122 - Part 1, Section 5. Most of what follows is based on the entities found in IGES:

5.6.X CONIC ARC

Parameters:

start point (P)
end point (p)
A,B,C,D,E,F (6R)

Description:

A conic arc is drawn which is defined as follows:

A conic arc is defined by the end points and the six parameters. The conic arc itself is defined by the six parameters in the following equation:

$$A*(XT**2) + B*XT*YT + C*(YT**2) + D*XT + E*YT + F = 0$$

where XT, YT is the plane of the definition space. In order for the conic arc to be processed correctly by the receiving system given the above representation, the conic arc entity must be positioned such that each of its axes is parallel to either the XT axis or YT axis. The arc is then positioned correctly in VDC space by using the value of the CONIC ARC TRANSFORMATION MATRIX element.

To determine the form of the conic arc, the quantities Q1, Q2 and Q3 are defined as follows:

$$Q1 = \text{determinant of} \begin{vmatrix} A & B/2 & D/2 \\ B/2 & C & E/2 \\ D/2 & E/2 & F \end{vmatrix}$$

$$Q2 = \text{determinant of} \begin{vmatrix} A & B/2 \\ B/2 & C \end{vmatrix}$$

$$Q3 = A + C$$

If Q2>0 and (Q1*Q3)<0, then the arc is an ellipse.
If Q2<0 and Q1<>0, then the arc is a hyperbola.
If Q2=0 and Q1<>0, then the arc is a parabola.

In the case where the conic arc is elliptical, to distinguish the arc in question from its compliment, the direction of the arc with respect to the definition space must be from start point to end point in a counterclockwise direction.

In the case where the conic arc is parabolic or hyperbolic, the parameterization defines a unique portion of the parabola or a unique portion of a branch of the hyperbola, thus the direction is irrelevant.

The direction of the conic arc with respect to VDC space is determined by the original direction of the arc in definition space, in conjunction with the action of the CONIC ARC TRANSFORMATION MATRIX element.

References:

5.6.X CONIC ARC TRANSFORMATION MATRIX

Parameters:

matrix elements

if the VDC type is 'integer',
R11,R12,R21,R22 (4I)

if the VDC type is 'real',
R11,R12,R21,R22 (4R)

coordinate offset (P)

Description:
This element is intended to work in conjunction with the CONIC ARC element to transform the conic arc from definition space to VDC space. The Transformation Matrix entity transforms the definition space point coordinates by means of a matrix multiplication and then the addition of an offset.

The notation for this transformation is as follows:

$$\begin{vmatrix} R11 & R12 \\ R21 & R22 \end{vmatrix} \begin{vmatrix} Xin \\ Yin \end{vmatrix} + \begin{vmatrix} T1 \\ T2 \end{vmatrix} = \begin{vmatrix} Xout \\ Yout \end{vmatrix}$$

where |Rij| is the transformation matrix, (Xin,Yin) is the coordinate to be transformed, (T1,T2) is the offset, and (Xout,Yout) is the coordinate resulting from the transformation. Both the input and output coordinate systems are assumed to be orthogonal, cartesian and right-handed.

References:

5.6.X PARAMETRIC SPLINE CURVE

Parameters:

curve_type (IX)
H-degree of continuity (I)
N-number of segments (I)
T-break point list for polynomial ((N+1)R)
X coordinate polynomial list (N sets of four)

AX,BX,CX,DX ((4*N)R)

Y coordinate polynomial list (N sets of four)

AY,BY,CY,DY ((4*N)R)

Description:
The parametric curve to be drawn is defined as follows:

The parametric spline curve is a sequence of parametric polynomial segments. The parameterization shown above is generalized to allow for the representation of many different parametric spline curves using this one element. The curve_type parameter indicates the type of parametric curve as it was represented in the sending system before being converted to this generic form. The following curve types have been assigned:

1: linear
2: quadratic

3: cubic
4: Wilson-Fowler
5: modified Wilson-Fowler
6: B spline

Values above 6 are reserved for registration and future standardization, and negative values are available for implementation-dependent use.

The degree of continuity parameter, H, indicates the smoothness, or continuity of the curve with respect to arc length. If H=0, the curve is continuous at all break points. If H=1, the curve is continuous and has slope continuity at all break points. If H=2, the curve is continuous and has both slope and curvature continuity at all break points.

The number of segments parameter, N, is the number of polynomial segments to be used to define the curve. Each X,Y polynomial segment is evaluated using the eight polynomial coefficients associated with that segment (AX,BX,CX,DX,AY,BY,CY,DY). Each segment is delimited by its respective breakpoint, T.

The following cubic polynomial equations will return the coordinates of the points of the i-th segment of the curve. Note that the coefficients D, or C and D will be zero if the polynomials are of degrees 2 or 1, respectively:

```
X(u) = AX(i) + BX(i)(s) + CX(i)(s**2) + DX(i)(s**3)
Y(u) = AY(i) + BY(i)(s) + CY(i)(s**2) + DY(i)(s**3)
```

where T(i) <= u <= T(i+1),i=1,...,N and s = u - T(i).

The terminate point and derivatives are derived without computing the polynomials by evaluating the Nth polynomials and derivatives at u = T(N + 1). These data, divided by the appropriate factorial (i.e. the second derivative divided by 2!, the third by 3!), are used as the N+1 or terminate point values.

References:

5.6.X RATIONAL B-SPLINE CURVE

Parameters:

K-upper index of sum (I)
M-degree of the basis function (I)
curve_open flag (one of: open, closed) (E)
equation_type flag (one of: rational, polynomial) (E)
periodic flag (one of: non-periodic, periodic) (E)
T-knot sequence list ((K+M+1)R)
W-weight list ((K+1)R)
control point list ((K+1)P)
start_param,end_param (2R)

Description:
A rational spline curve is drawn which is defined as follows:

The parametric equation governing the definition of the rational B-spline curve is shown in the following expression:

$$\frac{W(0)*P(0)*b0(t)}{W(0)*b0(t)} + \frac{W(1)*P(1)*b1(t)}{W(1)*b1(t)} + \ldots + \frac{W(K)*P(K)*bK(t)}{W(K)*bK(t)}$$

where W(i) are the weights, P(i) are the control points and bi are the basis functions.

The bi basis functions are all non-negative piecewise polynomials determined by the degree M, and the knot sequence T. T is a non-decreasing list of real numbers T(-M),...T(0),...T(K+1). Each function bi is supported by the knot sequence interval [T(i-M),T(i+1)]. Between any two adjacent knot values, the corresponding basis function can be expressed as a single polynomial of degree M.

The curve itself is parameterized where:

start_param <= t <= end_param,
T(0) <= start_param < end_param <= T(N)

Thus for any parameter value t between T(0) and T(K+1), the sum of the basis functions satisfies the following identity:

b0(t) + b1(t) + ... + bK(t) = 1.

If the beginning and ending points of the curve are identical, then the curve_open flag is set to closed, otherwise it is set to open.

If all of the weights in the weight list W are not equal, then the equation_type flag is set to rational. Otherwise, if all of the weights are equal, then all of the weights cancel, the denominators sum to one and the equation_type becomes polynomial.

References:

5.6.x COMPRESSED PEL ARRAY

Parameters:

T-encoding type (one of:T4,T6) (E)
P-pel path (one of:0,90,180,270) (E)
L-line progression (one of:90,270) (E)
S-pel spacing (R)
spacing_ratio (R)
N-number of pels per line (I)
NL-number of lines (I)
pel array (BS)

Description:
A compressed pel array image is defined as follows:

The encoding type parameter, T, specifies the CCITT compression format used to encode the image. If T is specified as "T4", the image is encoded according the one or two dimensional scheme defined in CCITT Recommendation T.4 (Group 3 facsimile). If T is "T6", the image is encoded according to the two dimensional scheme defined in CCITT Recommendation T.6 (Group 4 facsimile).

The pel path parameter, P, is the direction of progression of successive pels along a line relative to the VDC X axis. This parameter, in conjunction with the pel spacing, S, and the number of pels per line, N, implicitly define the line position, length and granularity for each

line in the decoded pel array. S is defined as the ratio m/n, where "m" is the line length in SMUs (1200 SMUs per inch) occupied by "n" pel spaces.

The line progression parameter, L, is the direction of progression of successive of pel lines and is expressed as a direction relative to P. L, in conjunction with the spacing_ratio and the number of lines, NL, implicitly defines the size of the decoded image in the direction of L. The line spacing (LS) of the lines of pels can be determined as follows:

LS = spacing_ratio * S

or rather, the spacing_ratio is the ratio of line spacing to pel spacing.

The pel array itself is stored in either of the formats defined by T, encoded as a bit_stream.

References:

5.6.x PEL ARRAY CLIP RECTANGLE

Parameters:

X1,Y1,X2,Y2 (4I)
first corner (P)
second corner (P)

Description:

The element defines the rectangular area of pels in the decoded pel array that is to appear in the picture.

The four integers form two coordinate pairs, (X1,Y1) and (X2,Y2) corresponding to the first and last pels to appear, respectively. For example, (6,2) would specify the seventh pel in line 3, given that (0,0) specifies the first pel on the first line. The default for the first coordinate is (0,0), and the default for the second is (N-1,L-1), where N is the number of pels per line and L is the number of lines from the COMPRESSED PEL ARRAY element.

The two corner points define the pel array clip rectangle in VDC space. The first pel defined above will appear at the first corner, and only those portions of the decoded pel array from the COMPRESSED PEL ARRAY element inside or on the boundary of the pel array clip rectangle will be drawn.

In addition to the proposed element listed and described on the preceeding pages, elements to add support for additional color modles (such as CIE, CMYB, HIS, etc.) is also under consideration. The form and content of these new (color modle related) elements is yet to be determined.

This page left intentionally blank.

APPENDIX 2

ADDENDUM 1 PDAD TEXT

ISO/TC97/SC24/N19

Information Processing Systems -

Computer Graphics -

Metafile for the Storage and Transfer of Picture Description Information

Part 1

Functional Specification

Addendum 1

November 1987

Page 1

Sub-clause 0.1: Add the following at the end of the sub-clause:

This picture description includes static images and graphical session capture requirements.

Page 1

Sub-clause 0.3: Add the following at the end of item c):

It should also not preclude further extensions to support future standards.

Page 1

Sub-clause 0.3: Add the following at the end of item d):

It should include the capability to support both GKS picture and graphical session requirements.

Page 3

Sub-clause 0.8: Add the following at the end of the first paragraph:

The CGM as extended by this addendum also specifies the elements required to support GKS picture and graphical session capture.

Page 4

Clause 1: Add the following at the end of the first paragraph:

This picture description includes static image and graphical session capture requirements.

Page 4

Clause 1: Add the following at the end of the second paragraph:

The CGM as extended by this addendum also contains elements that delimit and manipulate groups of elements within pictures. Capability is provided for dynamic picture regeneration such as is required for graphical session capture.

Page 9

Sub-clause 4.1: Add the following at the end of the list of classes of elements:

- Segment Elements, which enable the manipulation and appearance of elements within pictures. Segments can also appear outside pictures as global segments.

Page 9

Sub-clause 4.1: Add the following after the third paragraph:

Graphical output primitives and attributes may be grouped in segments. Segment attribute elements control the appearance of segments. Segments can appear both inside and outside pictures.

Page 10

Sub-clause 4.2: Add the following at the end of the sub-clause:

Groups of elements within pictures, called segments, are delimited by BEGIN SEGMENT and END SEGMENT. Each segment is uniquely identified by a segment identifier.

Add the following sub-clauses after sub-clause 4.3.2.2:

4.3.2.3 GKSM0 Set

The GKSM0 set includes all elements conforming to GKS level 0a in IS 7942.

The elements included in the GKSM0 set are:

BEGIN METAFILE
END METAFILE
BEGIN PICTURE
BEGIN PICTURE BODY
END PICTURE
METAFILE VERSION
METAFILE DESCRIPTION
VDC TYPE
INTEGER PRECISION
REAL PRECISION
INDEX PRECISION
COLOUR PRECISION
COLOUR INDEX PRECISION
MAXIMUM COLOUR INDEX
METAFILE ELEMENT LIST
METAFILE DEFAULTS REPLACEMENT
FONT LIST
CHARACTER SET LIST
CHARACTER CODING ANNOUNCER
VDC EXTENT
BACKGROUND COLOUR
MAXIMUM VDC EXTENT
VDC INTEGER PRECISION
VDC REAL PRECISION
CLIP RECTANGLE
MAKE PICTURE CURRENT
PREPARE VIEW SURFACE
UPDATE
DEVICE VIEWPORT
DEVICE VIEWPORT SPECIFICATION MODE
DEVICE VIEWPORT MAPPING
MODIFY FONT LIST
MODIFY CHARACTER SET LIST
POLYLINE
POLYMARKER
TEXT
POLYGON
CELL ARRAY
GDP
LINE BUNDLE INDEX
LINE TYPE
LINE WIDTH
LINE COLOUR
MARKER BUNDLE INDEX
MARKER TYPE
MARKER SIZE
MARKER COLOUR
TEXT BUNDLE INDEX
TEXT FONT
TEXT PRECISION
CHARACTER EXPANSION FACTOR
CHARACTER SPACING
TEXT COLOUR
TEXT PATH

TEXT ALIGNMENT
FILL BUNDLE INDEX
INTERIOR STYLE
FILL COLOUR
HATCH INDEX
PATTERN INDEX
FILL REFERENCE POINT
PATTERN TABLE
PATTERN SIZE
COLOUR TABLE
ASPECT SOURCE FLAGS
ESCAPE
APPLICATION DATA

Page 11

Sub-clause 4.3: Add the following sub-clause after sub-clause 4.3.3:

4.3.4 Metafile Categories

4.3.4.1 Introduction

Each metafile falls into a particular metafile category. The metafile category may be announced at the start of the metafile. This information may be used by the interpreter to decide if the metafile can be interpreted. The default metafile category is of the type 'cgm' as defined by IS 8632. The category implies that the metafile conforms to the semantics and formal grammar of that category. The metafile categories may overlap. The category does not imply particular default settings; these must be explicitly stated using the METAFILE DEFAULTS REPLACEMENT element.

4.3.4.2 CGMEXT1 Category

This category includes all elements defined for the 'cgm' category plus the elements defined for the first addendum.

4.3.4.3 GKSM Category

The GKSM category defines a metafile which conforms to the CGM standard and includes elements to support GKS, IS 7942, with appropriate requirements relating to position of elements in the metafile.

Page 12

Add the follwoing text to the end of sub-clause 4.4.4

MAXIMUM VDC EXTENT defines the space into which the VDC coordinate space is mapped for storage.

Page 15

Add the following sub-clauses after sub-clause 4.5.2:

4.5.3 Device Control

The extended CGM may contain information for the interpreter to use in controlling the output of the graphical information stored in the metafile.

4.5.4 Display Surface Concepts

Display surfaces are classified as being either hard-copy or soft-copy, on the basis of the medium that implements the surface. A surface that is hard copy is implemented by means of a medium that has to be replaced for each new image. One that is soft-copy is implemented by means of a medium that may automatically be cleared (typically my electrical means) for each new image.

Examples of hard-copy display surfaces are found in storage or refreshed cathode-ray tubes and in liquid crystal cells.

The PREPARE VIEW SURFACE function is used to ensure that the medium is ready to accept graphics at the start of a page or frame. It has a parameter that permits the client to specify that a hard-copy device should only advance the medium if it is known to have been marked upon (to conserve expensive media) or to force media advance (for example, to deliberatly create a blank frame or page in continuous media). This parameter is ignored by a soft-copy device, which always clears its display upon receipt of this command.

4.5.5 Deferral Mode Concepts

DEFERRAL MODE allows the storing on the metafile of information relating to the control of buffering and deferred actions of a graphics system. Deferral mode controls the possible delaying of output functions: for example, data sent to a device may be buffered to optimize data transfer. The values of deferral mode (in increasing order of delay) are:

a) ASAP: The visual effect of each function will be achieved As Soon As Possible (ASAP).

b) BNI: The visual effect of each function will be achieved Before the Next Interaction (BNI).

c) ASTI: The visual effect of each function will be achieved At Some Time (ASTI).

4.5.6 Device Viewport Control

The device viewport specifies the region of the actual device viewsurface into which the VDC extent is to be mapped.

The position of the device viewport is specified in one of three unit systems selected by device viewport specification mode parameters:

- by fraction [0.0 - 1.0] of the available display surface, which allows reasonable placement and relative sizing of the viewport, even without negotiation;
- in millimetres times a scale factor, which allows absolute sizing of images;
- in physical device units.

Another device viewport specification mode may be set to force isotropic mapping even if the specified VDC extent and device viewport would not otherwise have led to one. In such a case, the VDC extent is mapped onto a subset of the specified device vieport. This subset is defined by shrinking either the vertical or horizontal dimension of the specified viewport as needed to reach the required aspect ratio. This smaller "effective viewport" is then used to define the coordinate mapping from VDC to device coordinates. The placement of the effective viewport rectangle within the original one is specified by the third and fourth device viewport specification mode parameters: the options are left edge, right edge, or centred when the shrinking is horizontal, and top, bottom or centred when it is vertical. these meanings are relative to the non-inverted viewport.

The mapping of the VDC extent to the device can be mirrored in either the vertical or horizontal direction by reversing either the x or the y values in the device vieport specification. This implies that the images of all output primitives are inverted, including text and markers.

Page 17

Add the following at the end of Clause 4.6.3.2

The font list can be modified within a picture using the MODIFY FONT LIST element. This allows all or a part of the list to be modified.

The character set list can be modified in a similar way using the MODIFY CHARACTER SET LIST element.

Page 20

Add the following sub-clauses after sub-clause 4.6.7

4.6.8 Closed figures

The CGM as extended by this addendum provides the ability to construct a compound primitive to be filled, by a sequence of LINE and FILL AREA functions, EDGE attributes, and special control functions. This compound primitive or constructed region is referred to as a "closed figure", and is rendered using the same parity (odd/even) fill algorithm as used for other fill area functions.

The closed figure may be constructed of a combination of straight lines and curves and may contain "holes" as in POLYGON SET. In addition, the capability is provided to specify different portions of the edge using different EDGE attributes.

A closed figure is defined by the following functions:

control functions:	BEGIN FIGURE END FIGURE NEW REGION	MPLICIT EDGE VIS. ASF ESCAPE (Note 1)
boundary construction functions:	POLYLINE DISJOINT POLYLINE CIRC. ARC 3 PT. CIRC. ARC CTR. CIRC. ARC CTR. BACKWARDS ELLIPTICAL ARC GEN. DRAWING PRIM. (Note 2)	POLYGON POLYGON SET CIRC. ARC 3 PT. CLOSE CIRC. ARC CTR. CLOSE CIRCLE ELLIPTICAL ARC CLOSE ELLIPSE RECTANGLE
edge attribute functions:	EDGE COLOUR EDGE WIDTH EDGE TYPE	EDGE BUNDLE INDEX EDGE VISIBILITY

Note 1: Whether ESCAPE has meaning with regard to the construction of closed figures is dependent on the particular escape functionidentifier.

Note 2: Whether GENERALIZED DRAWING PRIMITIVE has meaning with regard to the construction of closed figures is dependent on the particular GDP function identifier.

4.6.9 Pixel Array

Pixel Array is similar to CELL ARRAY with the exception that no other mapping other than integer scaling of the array elements takes place. The colour information in the pixel array maps directly to the pixels of the target device. The starting point and colour information are specified in a device independent fashion, but the appearance of the final image depends directly on the resolution and aspect ratio of the target device as an MxN array of device independent colour specifiers that are assigned to an MxN array of pixels.

Page 22

Sub-clause 4.7: Add the following immediately above sub-clause 4.7.1:

The attribute elements LINE REPRESENTATION, MARKER REPRESENTATION, FILL REPRESENTATION, EDGE REPRESENTATION and TEXT REPRESENTATION are used to set all of the attribute values in a bundle table entry at the same time.

Page 40

Add the following sub-clauses after sub-clause 4.11:

4.12 Segment Elements

4.12.1 Introduction

In the CGM, graphical output primitives and attribute setting elements may be grouped in segments as well as being invoked outside segments. Each segment is identified by a unique segment identifier. Segments may be:

a. transformed;

b. made visible or invisible;

c. highlighted;

d. ordered front to back;

e. made detectable or undetectable;

f. deleted;

within a picture. They can also be defined as Global Segments as part of the Metafile Descriptor and can then be copied into the picture.

Only functions stored inside segments are affected by these operations.

Segments are the units for manipulation and change. Manipulation includes creation, deletion and renaming. Change includes transforming a segment, making a segment visible or invisible, highlighting a segment, and changing the order of overlapping segments.

The appearance of segments is controlled by segment attributes, which include segment transformation, visibility, highlighting, and segment display priority. Such segment attributes can be a basis for feedback during manipulations (for example, highlighting). The pick input properties of segments are also controlled by segment attributes, which include detectability and pick priority.

The segment elements are:

REOPEN SEGMENT
COPY SEGMENT
DELETE SEGMENT
DELETE ALL SEGMENTS
RENAME SEGMENT
REDRAW ALL SEGMENTS
IMPLICIT REGENERATION MODE
INHERITANCE FILTER
SEGMENT TRANSFORM
SEGMENT VISIBILITY
SEGMENT HIGHLIGHTING
SEGMENT DISPLAY PRIORITY
SEGMENT DETECTABILITY
SEGMENT PICK PRIORITY

4.12.2 Global and Local Segments

There are two types of segments: Local Segments and Global Segments. Both contain primitives and attributes which can be manipulated in the manner described above. Local Segments appear within a picture and are local to that picture. In contrast Global Segments can be used by a number of the pictures in the metafile in which they are defined.

4.12.2.1 Location of and Access to Global Segments.

A Global Segment is delimited by the normal BEGIN SEGMENT and END SEGMENT elements. Global Segments are defined in the Metafile Descriptor. They are not, by default, defined for or known to individual pictures in the metafile. They must are accessed from within individual pictures by the COPY function. The effect within a picture of a COPY function referring to a Global Segment is identical to the effect of a COPY function referring to a Local Segment.

4.12.2.2 Allowable Elements In MD and GSD States

BEGIN SEGMENT and END SEGMENT are the only segment-related elements that are allowed within the Metafile Descriptor (MD) state (see Table 3(a), the Metafile State Table). All of the segment attribute elements may occur within Global Segment Definition (GSD) state. The segment control elements REOPEN SEGMENT, DELETE ALL SEGMENTS, and REDRAW ALL SEGMENTS are not allowed in GSD state. Otherwise all of the segment control elements are allowed in GSD state (with the usual restriction that delete may not refer to the currently open segment). COPY is allowed in GSD state.

A number of other control elements are prohibited in GSD state. These are elements which make sense in Local Segment Definition (LSD) state, when a picture is open, but which would not make sense in GSD state, when the Metafile Descriptor is being processed. DEVICE VIEWPORT is an example of such a control element. These rules are concisely presented Table 3(a)

4.12.2.3 References to Global Segments

Within pictures, no elements are allowed that would modify the contents or default appearance of Global Segments. This includes all of the segment control elements and segment attributes. This restriction preserves logical independence of pictures and the ability to randomly access pictures. The only allowable references to Global Segments within pictures are via the COPY function.

4.12.2.4 Attribute Binding of Global Segments

Attributes are bound in Global Segments as they are in Local Segments. Upon the occurance of BEGIN METAFILE, every element that is modally defined and bound to primitives (Metafile Descriptor elements defining modes and precisions, Picture Descriptor elements, Control elements, and Attribute elements) has a well defined value. Conceptually the set of all of these define a "Modal State List".

The Metafile Descriptor is processed sequentially. Throughout the Metafile Descriptor, modal MD elements modify the MD entries in the state list and occurrances (possibly multiple) of the METAFILE DEFAULTS REPLACEMENT element allow manipulation (outside of GSD state) of all of the rest of the modal elements (as well as explicitly changing the defaults). Within GSD state the allowable modal elements (control, attribute, and segment attribute) also alter the contents of the Modal State List. The values of modal elements that are in effect upon BEGIN PICTURE are the default values, whether they are implicit (defined in the Standard) or explicit (i.e., via the Metafile Defaults Replacement).

4.12.3 Creating Segments

4.12.3.1 Segment Identifiers

Each segment has a unique identifier associated with it. Once a segment is deleted, its identifier is no longer associated with the segment and may be reused for another segment definition. The segment identifier associated with a closed segment may be changed with the RENAME SEGMENT function. Segment identifiers are of type SEGMENT NAME. The segment identifier is specified with the BEGIN SEGMENT function.

The supported range of segment identifiers is implementation-dependent. The segment identifier is used by BEGIN SEGMENT, REOPEN SEGMENT, COPY SEGMENT, DELETE SEGMENT, RENAME SEGMENT, REDRAW SEGMENT, and all segment attribute functions.

4.12.3.2 Opening and Closing Segments

The BEGIN SEGMENT function is the only means of creating a segment. Once a segment is opened, the current state of the individual primitive attributes, clipping rectangle, clip indicator, and pick identifier are known and the identifier of the open segment is set in the Segmentation State List. While a segment is open, the graphical objects passing along the CGI pipeline are stored in the segment. Segment attributes are also associated with the segment. Once open, the segment remains open for definition until END SEGMENT.

Segment attributes such as segment visibility, segment highlighting, and segment display priority affect the appearance of the display and may be set either while the segment is open or at any time after it is closed but not deleted.

A closed segment may be reopened at a later time with the REOPEN SEGMENT function. When a segment is reopened, graphical objects are stored in the segment using the same conceptual mechanism as when a segment is initially opened. When reopened, all subsequent graphical objects are appended to the open segment until END SEGMENT.

Consider the following example:

LINE COLOUR (blue)	
BEGIN SEGMENT (1)	
POLYLINE	blue lines
LINE COLOUR (yellow)	
POLYLINE	yellow lines
END SEGMENT	
POLYLINE	yellow lines
.	
.	
.	
LINE COLOUR (red)	
REOPEN SEGMENT (1)	reopen segment 1
POLYLINE	red lines
END SEGMENT	
POLYLINE	red lines

4.12.3.3 Graphical Objects

CGM elements may be grouped to conceptually form graphical objects. Such functions consist of primitives, attributes, and certain control capabilities. Refer to Table XX for a list of primitives, attributes, and control which conceptually may be used to form graphical objects.

The state list given later in this clause details those elements which are allowed in the segement open state for both local and global segments.

TABLE XX Functions used to form Graphical Objects

Primitives:	
POLYLINE	ELLIPTICAL ARC
DISJOINT POLYLINE	ELLIPTICAL ARC CLOSE
POLYMARKER	TEXT
POLYGON	APPEND TEXT
POLYGON SET	RESTRICTED TEXT
RECTANGLE	CELL ARRAY
CIRCLE	BEGIN FIGURE
ELLIPSE	END FIGURE
CIRC. ARC 3 POINT	NEW REGION
CIRC. ARC 3 POINT CLOSE	GENERALIZED DRAWING
CIRC. ARC CENTRE	PRIMITIVE (GDP)
CIRC. ARC CENTRE CLOSE	PIXEL ARRAY
CIRC. ARC CENTRE BACKWARDS	ESCAPE (Note 1)

Attributes and Control Capabilities:	
AUXILIARY COLOUR	EDGE BUNDLE INDEX
DRAWING MODE	EDGE TYPE
TRANSPARENCY	EDGE WIDTH
LINE BUNDLE INDEX	EDGE COLOUR
LINE TYPE	EDGE VISIBILITY
LINE WIDTH	IMPLICIT EDGE VISIBILITY
LINE COLOUR	CHAR. SET INDEX
MARKER BUNDLE INDEX	CHAR. EXPANSION FACTOR
MARKER TYPE	CHAR. CODING ANNOUNCER
MARKER SIZE	CHAR. SPACING
MARKER COLOUR	CHAR. HEIGHT
FILL BUNDLE INDEX	CHAR. ORIENTATION
FILL COLOUR	ALTERNATE CHAR. SET INDEX
FILL REFERENCE POINT	TEXT BUNDLE INDEX

INTERIOR STYLE
HATCH INDEX

PATTERN INDEX
PATTERN SIZE

TEXT PRECISION
TEXT COLOUR
TEXT PATH
TEXT ALIGNMENT
ASPECT SOURCE FLAGS
CLIP RECTANGLE
CLIP INDICATOR
PICK IDENTIFIER

NOTE: whether or not ESCAPE is a graphical object depends on the escape identifier, and is implementation-dependent.

4.12.3.4 Non-Retained Data

Any graphical object passing along the pipeline when no segment is open becomes non-retained data. The data is displayed in the usual manner. However, it is never redrawn as a result of implicit or explicit segment display. In particular, functions such as REDRAW ALL SEGMENTS will not redraw data which has not been stored in a segment.

4.12.4 Segment Attributes

4.12.4.1 Introduction

The segment attributes associated with each segment control display and pick input properties. Segment attributes can be set either while the segment is open or any time after it is closed but not deleted. When a segment is opened with the BEGIN SEGMENT function, the segment's attributes are set to their default values.

Each of the segment attributes are discussed in detail in the remaining parts of this section.

Segment attributes are associated with the segment rather than the segment identifier. This means that all segment attributes no longer exist for the segment when it is deleted. This also means that the segment attributes are not changed when a segment is renamed.

4.12.4.2 Segment Highlighting

Segment highlighting has two states, NORMAL and HIGHLIGHTED. The setting of this attribute selects one of these two states for the segment. The nature in which highlighting is represented is implementation-dependent. When a segment's highlighting attribute is changed, all of the graphical objects of the segment are displayed based on the implicit segment regeneration mode and the segment display priority.

4.12.4.3 Segment Visibility

Segment visibility can be set to VISIBLE or INVISIBLE. When a segment is set to be VISIBLE, all of the graphical objects are displayed based upon the implicit segment regeneration mode and the segment display priority. A segment can be picked only if it is both VISIBLE and DETECTABLE.

4.12.4.4 Segment Detectability

A segment can be set to be DETECTABLE or UNDETECTABLE. A segment can be picked only if it is both VISIBLE and DETECTABLE. Segment detectability does not affect the display or appearance of segments.

4.12.4.5 Segment Display Priority

The segment display priority attribute determines how overlapping segments are displayed. In general, segments with higher display priorities will be displayed as if they were in front of segments with lower display priorities.

The SEGMENT DISPLAY PRIORITY function is used to change the display priority of a segment. When a segment's display priority is changed, the segment's display is based on the implicit segment regeneration mode and the new display priority.

4.12.4.6 Segment Pick Priority

Each segment has an associated pick priority. The pick priority is used to resolve the picking of segments which overlap. If two or more segments overlap and the pick location is within the intersection of these segments, the segments picked will be one or more of those with the highest pick priority. If more than one of the overlapping segments has the highest pick priority, then the segments picked will be the segments with both the highest pick priority and highest and display priority. In such cases a list of segment identifiers is returned with the identical and highest pick priorities and display priorities.

4.12.4.7 Segment Transform

The segment transform is a coordinate transform associated with each segment. It allows scaling, translation, and rotation of segments.

Note that the segment transform is distinct from the VDC EXTENT/DEVICE VIEWPORT mapping, which is a transform of VDC space to DC space. The segment transform is a transform of VDC space to VDC space.

The segment transform is applied to reference points and parameters and attributes with significance in VDC space. The reference points are defined to be all input parameters of type POINT. Parameters with significance in VDC space include the radius of a CIRCLE. Attributes with significance in VDC space include scalars such as PATTERN SIZE, CHARACTER ORIENTATION, and LINE WIDTH, EDGE WIDTH and MARKER SIZE when they are specified in VDC. Other attributes which may be transformed include CHARACTER HEIGHT and FILL REFERENCE POINT.

Transformation of the radius of a CIRCLE is evaluated by transforming a radius vector aligned in an implementation- dependent manner. Similarly, if a rotation transform were applied to a RECTANGLE, the corners of the RECTANGLE will change, but the edges of the rectangle will remain horizontal and vertical. If a scaling transform is applied to a PIXEL ARRAY, the location of the PIXEL ARRAY changes, but not its size. If a transform is applied to RESTRICTED TEXT, its reference points are changed. That is, its location and its extent will change.

When the mapping of the VDC space to device or visual space is anisotropic, due either to a segment transform or the VDC EXTENT, DEVICE VIEWPORT, and DEVICE VIEWPORT MAPPING, thick lines may be rendered in either of two ways:

- A uniform transformation may be applied, such that the thick line behaves as would a polygon equivalent to the two-dimensional realization of the thick line. In this case, the rendered width of the line will change with the direction of the line segment.

- A fixed scaling may be selected which is between the minimum (scale in X, scale in Y) and maximum (scale in X, scale in Y), and applied to all line segments independent of their direction.

The rendering of markers is intended not to be transformed by either anisotropic mapping established by VDC EXTENT and related commands, nor by the segment transform. That is, the shape of the marker symbol is not to be altered.

When MARKER SIZE is specified in VDC units, the MARKER SIZE is converted into a vector and transformed; a single scale factor between the minimum (scale in X , scale in Y) and maximum (scale in X, scale in Y) is selected and applied uniformly.

A segment transform is represented by a 2x3 matrix, composed of a 2x2 scaling and rotation portion, and a 2x1 translation portion. The default segment transform is represented by the identity transform. If the user never invokes the SEGMENT TRANSFORM function, then all coordinate data is mapped the same as non-retained data (i.e., the VDC EXTENT/DEVICE VIEWPORT is the only mapping). If the user alters the segment's transform attribute, each of the reference points discussed previously will be transformed by the segment transform, producing new data.

The segment transform is a segment attribute. This transform may be modified by the SEGMENT TRANSFORM function. The stored transform is applied each time the segment is redrawn. Setting the transform back to identity will produce the original picture with no loss of data on the next redraw.

The use of segment transforms may produce coordinates that cannot be expressed within the VDC range. This is handled in an implementation-dependent way.

The sequence of transforms is optionally the copy transform followed by the segment transform, then clip to a clipping area, followed by the VDC EXTENT/DEVICE VIEWPORT transform. For non-retained data there is no segment transform. For data stored in segments, the segment transform maps VDC to VDC. Clipping rectangles are not transformed by the segment transform. The last transform, VDC EXTENT/DEVICE VIEWPORT, is finally applied to map VDC to DC.

4.12.5 Segment Display

4.12.5.1 Introduction

Segment display is the process which produces a visible image on a display surface from the graphical objects in a segment. Segments can always be displayed, individually or collectively, by the explicit invocation of REDRAW SEGMENT, or REDRAW ALL SEGMENTS. Segments can also be displayed implicitly, under some circumstances, without using the above functions.

Several of the segment attributes affect segment display. Segment highlighting determines how a segment is displayed. It can be set to NORMAL or HIGHLIGHTED. Segment visibility determines whether the graphical objects within a segment are displayable. It can be set to VISIBLE or INVISIBLE. Segment display priority determines which overlapping segments appear to be in front. Segment transform affects the position, size, and orientation of the displayed segment. Segment detectability is a segment attribute that affects the capability of a segment to be picked.

Primitive attributes which are conceptually part of the graphical object's definition also affect the appearance of the displayed segment.

4.12.5.2 Segment Regeneration

Implicit segment regeneration mode determines when picture changes occur. Operations not immediately supported by devices may require certain functions to perform an implicit regeneration. For example, it is possible to create a segment and then modify one of the segment attributes which could cause the picture to change. Some devices such as plotters are only capable of doing additive output, as is the case with no segmentation. For example, if such a device received a function to change the segment transform, then it would have to advance the paper or film and regenerate the entire picture. In order to prevent the waste of time and paper, the CGM as extended by this addendum has an IMPLICIT SEGMENT REGENERATION MODE function.

The IMPLICIT SEGMENT REGENERATION MODE function may be used to stop such implicit actions from occurring immediately. If the mode is set to SUPPRESSED then no implicit regeneration occurs. It is then up to the user to decide when the picture is complete. At that time the user explicitly invokes the PREPARE VIEW SURFACE and REDRAW ALL SEGMENTS functions.

The implicit segment regeneration mode should be SUPPRESSED on devices which must perform a regeneration to display the result of segment changes. The default setting of implicit regeneration mode is SUPPRESSED.

It should be noted that not all functions cause implicit regeneration. The actions which may cause implicit regeneration include changing segment attributes such as segment display priority, segment visibility, segment highlighting and segment transform. In addition, if a segment is open and the implicit segment regeneration mode is set to ALLOWED, regeneration may take place as graphical objects are added to the segment. The implicit segment regeneration mode merely suppresses or allows regeneration.

Setting the mode to SUPPRESSED does not mandate that picture changes are suppressed. It merely stops devices from regenerating the picture if the device uses implicit regeneration as the method for implementing some CGM elements. If the intention is to delay the visual effect of output then the SEGMENT VISIBILITY function should be used. Implicit segment regeneration should not be confused with deferral mode. Segment regeneration takes precedence over the deferral mode.

4.12.5.3 Implicit Segment Display

When implicit segment regeneration mode is ALLOWED, segments may be regenerated implicitly. Implicit segment regeneration may happen initially as the segment is being defined. Closed segments may also be regenerated implicitly to maintain a current image on the display surface. In this latter case, implicit regeneration would typically happen when interpretation of a CGM element changes the appearance of the displayed image in some respect. For example, implicit regeneration may be required when changing the

display priority of a segment. It may also be required when changing the segment transform, segment visibility, and segment highlighting attributes.

4.12.5.4 Explicit Segment Display

The elements REDRAW SEGMENT and REDRAW ALL SEGMENTS can be used explicitly to display visible segments, independent of the implicit segment regeneration mode. REDRAW ALL SEGMENTS displays all defined visible segments without clearing the display surface. REDRAW SEGMENT draws the segment identified. It is implementation-dependent whether segments that overlap the identified segment are also redrawn with the REDRAW SEGMENT function. REDRAW SEGMENT and REDRAW ALL SEGMENTS are most useful when the implicit segment regeneration mode is set to SUPPRESSED.

4.12.6 Copy Segment and the Inheritance Filter

The COPY SEGMENT function copies the graphical objects in the identified segment into the open segment, applying a coordinate transformation as the objects are copied.

The objects copied may be altered in a variety of ways:

a. The inheritance filter controls whether individual attribute values are reapplied to the graphical objects

b. The graphical objects are transformed by the copy segment transformation according to the rules for transformation

An example of the COPY SEGMENT function is as follows:

LINE COLOUR (blue)	
BEGIN SEGMENT (1)	
LINE STYLE (dotted)	
POLYLINE	blue dotted line
END SEGMENT	
LINE COLOUR (red)	
LINE STYLE (dashed)	
BEGIN SEGMENT (2)	
POLYLINE	red dashed line
COPY SEGMENT (1)	the copy of segment (1) POLYLINE still generates a blue dotted line, unless changed with the inheritance filter.

The CGM generator has the capability of setting a copy transform. The copy segment transform is applied to graphical objects before they are copied into the open segment. However, this does not apply to clipping rectangles. Graphical objects may be transformed to alter the location, the size, and the orientation of primitives.

An example of the COPY SEGMENT function with the INHERITANCE FILTER function is as follows:

INHERITANCE FILTER (LINE ATTRIBUTES, STATE_LIST)	
BEGIN SEGMENT (1)	
LINE COLOUR (blue)	
LINE STYLE (dotted)	
POLYLINE	blue dotted line
END SEGMENT	
LINE STYLE (dashed)	
LINE COLOUR (red)	
BEGIN SEGMENT (2)	
POLYLINE	red dashed line
COPY SEGMENT (1)	red dashed line since the inheritance filter selects STATE_LIST for line attributes.

4.12.7 Delete and Rename Segments

The DELETE SEGMENT function is used to delete an individual segment. Likewise, the DELETE ALL SEGMENTS function deletes all segments. Segment identifiers for segments which have been deleted are immediately available for reuse.

If an attempt is made to delete the open segment, either by DELETE SEGMENT or by DELETE ALL SEGMENTS, an error is detected and the function is ignored.

When a segment is deleted it is erased from the display. This action might be delayed by the implicit segment regeneration mode. Deleted segments may be erased by not redrawing them at the next regeneration. Deleting a segment can cause an implicit regeneration if such is allowed.

An existing segment may be renamed at any time, except when open. (When a segment is renamed it is immediately associated with a new segment identifier.) The segment's old identifier is immediately available for reuse by BEGIN SEGMENT.

4.12.8 Save, Restore and Delete Attributes

Three functions are provided to save and restore attributes. This capability allows the client to preserve the state of all primitive attributes, excluding bundle tables, character set list, font list, colour table, and pattern table in a 'named' area. This capability can be used to save and restore attributes in conjunction with opening and closing segments.

Page 41

Add the following to Figure 12:

figure 12 modifications to add segments

Page 41

Add the following table following the state diagram

Table 3(a) CGM Elements by their allowed states

CGEM Element	CGEM states PC	MD	GS	PD	PO	TO	LS	FC	FO
BEGIN PICTURE	X	X							
BEGIN PICTURE BODY				X					
END PICTURE					X				
BEGIN SEGMENT		X			X				
END SEGMENT			X				X		
END METAFILE	X								
METAFILE VERSION		X							
METAFILE DESCRIPTION		X							
VDC TYPE		X							
INTEGER PRECISION		X							
REAL PRECISION		X							
INDEX PRECISION		X							
COLOUR PRECISION		X							
COLOUR INDEX PRECISION		X							
MAXIUMUM COLOUR INDEX		X							
COLOUR VALUE EXTENT		X							

METAFILE ELEMENT LIST	X						
METAFILE DEFAULTS REPL.	X						
FONT LIST	X						
CHARACTER SET LIST	X						
CHAR CODING ANNOUNCER	X						
METAFILE CATEGORY	X						
MAXIMUM VDC EXTENT	X						
SEGMENT PRIORITY EXTENT	X						
SCALING MODE		X					
COLOUR SELECTION MODE		X					
LINE WIDTH SPEC MODE		X					
MARKER SIZE MODE		X					
EDGE WIDTH SPEC MODE		X					
VDC EXTENT		X	X		X		
BACKGROUND COLOUR		X					
VDC INTEGER PRECISION	X		X		X	X	X
VDC REAL PRECISION	X		X		X	X	X
AUXILIARY COLOUR	X		X	X	X	X	X
TRANSPARENCY	X		X	X	X	X	X
CLIP RECTANGLE	X		X		X	X	X
CLIP INDICATOR	X		X		X	X	X
DEVICE VIEWPORT			X		X		
DEVICE VIEWPORT MAPPING			X		X		
DEVICE VPORT SPEC MODE			X		X	X	X
DEFERRAL MODE			X		X	X	X
MAKE PICTURE CURRENT			X		X	X	X
PREPARE VIEW SURFACE			X		X	X	X
UPDATE			X		X	X	X
MODIFY FONT LIST			X		X		
MODIFY CHARACTER SET LIST			X		X		
BEGIN FIGURE	X		X		X		
END FIGURE	X		X		X	X	X
NEW REGION	X		X		X		X
IMPLICIT EDGE VISIBILITY	X		X		X	X	X
SAVE PRIMITIVE ATTRIBUTES	X		X		X		
RESTORE PRIMITIVE ATTRIBS	X		X		X		
DELETE PRIMITIVE ATT SET	X		X		X		
POLYLINE	X		X		X	X	X
DISJOINT POLYLINE	X		X		X	X	X
POLYMARKER	X		X		X		
TEXT	X		X		X		
RESTRICTED TEXT	X		X		X		
APPEND TEXT	X		X	X	X		
POLYGON	X		X		X	X	
POLYGON SET	X		X		X	X	
CELL ARRAY	X		X		X		
GDP	X		X		X	X	X
RECTANGLE	X		X		X	X	
CIRCLE	X		X		X	X	
CIRCLAR ARC 3 POINT	X		X		X	X	X
CIRCULAR ARC 3 PT CLOSE	X		X		X	X	
CIRCULAR ARC CENTRE	X		X		X	X	X
CIRC ARC CENTRE CLOSE	X		X		X	X	
ELLIPSE	X		X		X	X	
ELLIPTICAL ARC	X		X		X	X	X
ELLIPTICAL ARC CLOSE	X		X		X	X	
CIRC ARC CENTRE BKWDS	X		X		X	X	X
PIXEL ARRAY	X		X		X		
LINE BUNDLE INDEX	X		X		X		
LINE TYPE	X		X		X		

LINE WIDTH			X		X		X		
LINE COLOUR			X		X		X		
MARKER BUNDLE INDEX			X		X		X		
MARKER TYPE			X		X		X		
MARKER SIZE			X		X		X		
MARKER COLOUR			X		X		X		
TEXT BUNDLE INDEX			X		X	X	X		
TEXT FONT INDEX			X		X	X	X		
TEXT PRECISION			X		X	X	X		
CHAR EXPANSION FACTOR			X		X	X	X		
CHARACTER SPACING			X		X	X	X		
TEXT COLOUR			X		X	X	X		
CHARACTER HEIGHT			X		X	X	X		
CHARACTER ORIENTATION			X		X		X		
TEXT PATH			X		X		X		
TEXT ALIGNMENT			X		X		X		
CHARACTER SET INDEX			X		X	X	X		
ALT CHAR SET INDEX			X		X	X	X		
FILL BUNDLE INDEX			X		X		X		
INTERIOR STYLE			X		X		X		
FILL COLOUR			X		X		X		
HATCH INDEX			X		X		X		
PATTERN INDEX			X		X		X		
EDGE BUNDLE INDEX			X		X		X	X	X
EDGE TYPE			X		X		X	X	X
EDGE WIDTH			X		X		X	X	X
EDGE COLOUR			X		X		X	X	X
EDGE VISIBILITY			X		X		X	X	X
FILL REFERENCE POINT			X		X		X		
PATTERN TABLE			X		X		X		
PATTERN SIZE			X		X		X		
COLOUR TABLE			X		X		X		
ASPECT SOURCE FLAGS			X		X		X	X	X
LINE REPRESENTATION			X		X		X		
MARKER REPRESENTATION			X		X		X		
TEXT REPRESENTATION			X		X		X		
FILL REPRESENTATION			X		X		X		
EDGE REPRESENTATION			X		X		X		
DRWAING MODE			X		X		X		
ESCAPE	X	X	X	X	X	X	X	X	X
MESSAGE	X	X	X	X	X	X	X	X	X
APPLICATION DATA	X	X	X	X	X		X	X	X
REOPEN SEGMENT					X				
RENAME SEGMENT			X		X		X		
DELETE SEGMENT			X		X		X		
DELETE ALL SEGMENTS					X				
REDRAW ALL SEGMENTS					X		X		
COPY SEGMENT			X		X		X		
IMPLICIT SEG REGEN MODE			X		X		X		
INHERITANCE FILTER			X		X		X		
SEGMENT TRANSFORM			X		X		X		
SEGMENT VISIBILITY			X		X		X		
SEGMENT HIGHLIGHTING			X		X		X		
SEGMENT DISPLAY PRIORITY			X		X		X		
SEGMENT PICK PRIORITY			X		X		X		
SEGMENT DETECTABILITY			X		X		X		

PC Picture Closed
MD Metafile Descriptor

GS	Global Segment
PD	Picture Descriptor
PO	Picture Open
TO	Text Open
LS	Local Segment
FC	Figure Closed
FO	Figure Open

Page 42

Sub-clause 5.1: Add the following after the ninth paragraph which starts with the sentence: "The External Elements....":

The Segment Elements (described in sub-clause 5.10) provide for the grouping and manipulation of elements.

Page 42

Sub-clause 5.1: Add the following at the end of the table of abbreviations of data type names:

N	Name	Identifier of type Integer
DP	Device Point	A point expressed in a coordinate system that is device dependent. DP units are metres or other appropriate device units.
ASN	Attribute Set Name	identification of a set of values used with SAVE and RESTORE PRIMITIVE ATTRIBUTES
SN	Segment Name	Segment Identifier Realization is encoding dependent. Range is implementation dep.
VSP	Viewport Specification Point	Two values representing the x and y coordinates of a point in viewport specification space, whose data type and interpretation depend on DEVICE VIWPORT SPECIFICATION MODE.

Page 46

Add the following sub-clauses after sub-clause 5.2.5:

5.2.6 BEGIN SEGMENT

Parameters:

Segment Identifier (SN)

Description:

This is the first element of a segment. All subsequent elements until the next END SEGMENT will be collected into this segment.

5.2.7 END SEGMENT

Parameters:

None

Description:

Subsequent elements will no longer be part of a segment.

Page 47

Add the following at the end of the Description section of sub-clause 5.3.1

The CGM as extended by Addendum 1 is version two (2).

Page 50

Sub-clause 5.3.11: Add the following shorthand name at the end of the list given in the second paragraph of the "Description":

GKSM0

Page 51

Add the following note at the end of sub-clause 5.3.3: FONT LIST

NOTE: It is recommended that a dense font list is used.

Page 55

Add the following sub-clauses after sub-clause 5.3.15:

5.3.16 METAFILE CATEGORY

Parameters:

category (one of: cgm, gksm, cgmext1) (E)

Description:

This function sets the metafile category to the type indicated by the parameter.

5.3.17 MAXIMUM VDC EXTENT

Parameters:

low value	(Point)
high value	(Point)

Description:

The parameters define a mapping of a sub-space of the VDC range defined by (low,low) and (high,high) and the virtual coordinate space of a graphics system, e.g. NDC, such that the (low,low) corner is equivalent to the lower left corner of NDC, and the (high,high) corner with the upper right corner of NDC. The low value is less than the high value.

5.3.18 SEGMENT PRIORITY EXTENT

Parameters:

minimum extent	(integer)
maximum extent	(integer)

Description:

The parameters represent an extent which bounds the segment display priority values which will be found in the metafile. It need not represent the exact extent of the values contained in the metafile.

Add the following sub-clauses after sub-clause 5.5.6:

5.5.7 DEVICE VIEWPORT

Parameters:

first corner	(DP)
second corner	(DP)

Description:

The two parameters define the opposite corners of a rectangular viewport on the device's display surface.

5.5.8 DEVICE VIEWPORT SPECIFICATION MODE

Parameters:

VSU specifier	(one of: fraction of default device viewport millimetres with scalefactor, physical device units)(E)
Metric scale factor	(R)

Description

This function determines how subsequent functions using the data type VSU (viewport specification unit) or VSP (viewport specification point) will be defined. (This applies primarily, but not exclusively to the DEVICE VIEWPORT function.)

These parameters may be specified in one of three systems of units.

When the first parameter is 'fraction of default device viewport ', the value (0.0, 0.0) corresponds to the lower left corner and the value (1.0, 1.0) corresponds to the upper right corner of the default viewport. (The default device viewport is the largest unrotated rectangular area visible on the display surface.) Numbers outside of the range [0.0..1.0] may be specified (see DEVICE VIEWPORT). The second parameter is ignored.

When the parameter is 'millimetres with scalefactor', the metric scale factor parameter represents the distance (in millimetres) on the view surface corresponding to one VSU. One VSU represents one millimetre multiplied by the metric scale factor. The value (0,0) corresponds to the lower left corner and the values increase positively to the right and upwards.

When the parameter is 'physical device units', the native units and handedness of the physical device are used. The second parameter is ignored.

Note - Metric scaling with a scale factor provides a device-independent means of generating output at a known scale factor. In metric mode, a scale factor of 1.0 indicates that the VSU are in units of millimetres; a scale factor of 0.0254 would imply VSU of one thousand per inch.

Note - The only allowed data type for physical device units is integer. Physical devices which support real number physical addressing must be accessed through numeric type translation.

5.5.9 DEVICE VIEWPORT MAPPING

Parameters:

Isotropy flag	(one of: not forced, forced)(E)
Horizontal alignment flag	(one of: Left, Centre, Right)(E)
Vertical alignment flag	(one of: Bottom, Centre, Top)(E).

Description:

This function determines how the coordinate mapping is derived from the VDC EXTENT and the specified DEVICE VIEWPORT. See the DEVICE VIEWPORT description for a fuller explanation. The remaining parameters are only significant if isotropy is forced by the first parameter. If so, the

effective viewport is generally smaller than the specified viewport, and these parameters determine how it will be positioned within the specified viewport. 'Left' and 'bottom' are interpreted as being towards the 'first corner' of the specified DEVICE VIEWPORT, regardless of any mirroring or rotation of the viewport on the physical device.

5.5.10 DEFERRAL MODE

Parameters:

deferral mode (one of: asap, bnig, bnil, asti) (E)

Description:

Deferral mode controls the possible delaying of output functions: for example, data sent to a device may be buffered to optimize data transfer. The values of deferral mode (in increasing order of delay) are:

a) ASAP: The visual effect of each function will be achieved As Soon As Possible (ASAP).

b) BNI: The visual effect of each function will be achieved Before Next Interaction (BNI)

d) ASTI: the visual effect of each function will be achieved At Some Time (ASTI).

5.5.11 MAKE PICTURE CURRENT

Parameters:

None

Description:

This function ensures that any buffered data is sent to the device's display surface, such that it is visible to the viewer.

5.5.12 PREPARE VIEW SURFACE

Parameters:

Force hardcopy advance (one of: force hardcopy, conditional)(E)

Description:

This function discards any pending data, clears the display surface of a softcopy device, and clears any internally-stored display list. The cleared region is set to the colour specified by BACKGROUND COLOUR, if background colour is supported.

For a hardcopy device, if 'force hardcopy advance' is set to 'force hardcopy', the medium is advanced unconditionally. If the parameter is set to 'conditional', the medium is not advanced if it is known that it has not been marked; if the medium is marked, or if the implementation cannot tell whether it has been marked, then the medium is advanced. The parameter is ignored by a softcopy device.

5.5.13 UPDATE

Parameters:

update regeneration flag (one of: perform, postpone) (E)

Description:

This element indicates that all deferred actions are executed when the metafile is interpreted (without intermediate clearing of the display surface).

5.5.14 MODIFY FONT LIST

Parameters:

starting index (I)
font list (S)

Description:

This element modifies the font list specified in the FONT LIST element or the default list. Only the specified entries in the font list are modified.

Legal values of the starting index are non-negative integers.

5.5.15 MODIFY CHARACTER SET LIST

Parameters:

starting index (I)
list of:
character set type - one of: 94-character G-set,
96-character G-set,
94-character multibyte G-set,
96-character multibyte G-set,
complete code) (E),
designation sequence tail (S)

Description:

This element modifies the list of Character Sets given in the Metafile Descriptor via the CHARACTER SET LIST element or the default list. Only the specified entries are modified.

5.5.16 BEGIN FIGURE

Parameters:

none

Description:

The metafile enters the state 'FIGURE OPEN REGION CLOSED', initiating construction of the compound closed figure primitive.

The receipt of a matching END FIGURE function signals the rendering of the closed figure, and a transition out of either 'FIGURE OPEN REGION CLOSED' or 'FIGURE OPEN REGION OPEN' state back to 'ACTIVE'.

If the metafile is already in either of the "figure open" states, the previous portion of the closed figure boundary definition is discarded, and the metafile remains in the "figure open" state. That is, the effect is as if the boundary definition was started anew.

5.5.17 END FIGURE

Parameters:

none

Description:

This function causes transition out of either the 'FIGURE OPEN REGION OPEN' or 'FIGURE OPEN REGION CLOSED' state, and causes the compound closed figure primitive to be rendered.

If the metafile was in state 'FIGURE OPEN REGION OPEN', and the "last point" of the last LINE function is not coincident with the current closure point, then the closed figure has not been explicitly closed and an implicit closure is performed by connecting the last point of the preceding LINE function to the current closure point. The visibility of this line segment is controlled by IMPLICIT EDGE VISIBILITY.

5.5.18 NEW REGION

Parameters:

none

Description:

This function is used only within the 'FIGURE OPEN REGION OPEN' state, for control of subregion construction within closed figures.

If the current region has not yet been closed by a preceding NEW REGION function, and the last point of the last LINE function is not coincident with the current closure point, then the current subregion is closed by a line segment connecting the last point of the preceding LINE function to the current closure point. If the region has been previously closed (i.e., the metafile is in state 'FIGURE OPEN REGION CLOSED'), is empty, or the last point of the last LINE function is coincident with the current closure point, then no line segment is generated by this function.

The visibility of this line segment is controlled by IMPLICIT EDGE VISIBILITY.

The metafile goes to the state 'FIGURE OPEN REGION CLOSED'. The first point of the next LINE function following a CLOSE REGION function becomes the new closure point, starting a new subregion.

5.5.19 IMPLICIT EDGE VISIBILITY

Parameters:

implicit edge visibility (one of: off,on)(E)

Description:

IMPLICIT EDGE VISIBILITY specifies whether edges added implicitly during the construction of a closed figure are to be rendered as visible or invisible. It provides control for each edge, that is, each implicitly added edge obeys the current value of IMPLICIT EDGE VISIBILITY.

Edges are added implicitly when the following situations arise in the "figure open" state:

1. The "first point" of a LINE function is not coincident with the "last point" of the preceding LINE function.

2. The DISJOINT POLYLINE function is used.

3. A NEW REGION function is invoked, the metafile is in state 'FIGURE OPEN REGION OPEN', and the "last point" of the preceding LINE function is not coincident with the current closure point.

4. An END FIGURE function is invoked, the metafile is in state 'FIGURE OPEN REGION OPEN', and the "last point" of the preceding LINE function is not coincident with the current closure point.

In each of these cases, the edge connecting the two points is added to the boundary definition.

This function may be used anywhere within a picture (i.e., both before and during the closed figure definition); however, it only affects the rendering of closed figures.

5.5.20 SAVE PRIMITIVE ATTRIBUTES

Parameters:

Attribute set name ASN

Description:

This function allows the generator to save under the given name a copy of the current state list entries for those attributes and controls listed in the following table.

If the attribute set name is already in use, the values saved under that name are replaced by the current values and no error is generated.

TABLE XX Functions whose state list entries are saved by SAVE PRIMITIVE ATTRIBUTES and restored by RESTORE PRIMITIVE ATTRIBUTES

AUXILIARY COLOUR (Note 1)
DRAWING MODE
TRANSPARENCY

LINE BUNDLE INDEX
LINE TYPE
LINE WIDTH (Note 2)
LINE COLOUR (Note 1)

MARKER BUNDLE INDEX
MARKER TYPE
MARKER SIZE (Note 2)
MARKER COLOUR (Note 1)

FILL BUNDLE INDEX
FILL COLOUR (Note 1)
FILL REFERENCE POINT
INTERIOR STYLE
HATCH INDEX

PATTERN INDEX
PATTERN SIZE

EDGE BUNDLE INDEX
EDGE TYPE
EDGE WIDTH (Note 2)
EDGE COLOUR (Note 1)
EDGE VISIBILITY

IMPLICIT EDGE VISIBILITY

CHARACTER SET INDEX
CHARACTER EXPANSION FACTOR
CHARACTER CODING ANNOUNCER
CHARACTER SPACING
CHARACTER HEIGHT
CHARACTER ORIENTATION
ALTERNATE CHARACTER SET INDEX
TEXT BUNDLE INDEX
TEXT PRECISION
TEXT COLOUR (Note 1)
TEXT PATH
TEXT ALIGNMENT

ASPECT SOURCE FLAGS

CLIP RECTANGLE
CLIP INDICATOR

Note 1: The COLOUR SELECTION MODE in which this value was last set is also stored.

Note 2: The corresponding specification mode in which this value was last set is also stored.

5.5.21 RESTORE PRIMITIVE ATTRIBUTES

Parameters:

Attribute set name ASN

Description

The values saved by the last SAVE PRIMITIVE ATTRIBUTES using the given attribute set name are copied into the current State List being used on interpretation.

5.5.22 DELETE ATTRIBUTE SET

Parameters:

Attribute set name ASN

Description

The previously saved attribute set is deleted from the sets stored.

Page 78

Add the following sub-clauses after sub-clause 5.6.19

5.6.20 CIRCULAR ARC CENTRE BACKWARDS

Parameters:

centrepoint (P)

DX_start, DY_start, DX_end, DY_end (4VDC)
radius (VDC)

Description:

A circular arc is drawn which is defined as follows:

DX_start and DY_start define a start vector, and DX_end and DY_end define an end vector. The tails of these vectors are placed on the centrepoint. A start ray and end ray are derived from the start and end vectors. The start and end rays are semi-infinite lines from the centrepoint in the directions of the start and end vectors respectively.

The specified radius and centrepoint define a circle. The arc is drawn in the negative angular direction (as defined by VDC EXTENT) from the intersection of the circle and the start ray (as obtained by measuring a distance 'radius' along the start ray from the centrepoint) to the intersection of the circle and the end ray.

The arc is displayed with current line element attributes.

Valid values of the vector components are those which produce vectors of non-zero length.

Valid values of 'radius are non-negative VDC.

If the start ray and end ray are coincident, it is ambiguous whether the defined arc subtends 0 degrees or 360 degrees of central angle (see also annex D).

5.6.21 PIXEL ARRAY

Parameters:

origin point	(P)
nx,ny	(2I)
valid x range	(2I)
valid y range	(2I)
x scale	(I)
y scale	(I)
colour specifiers	(nx*ny CO)

Description:

Assigns a row major rectangular array of device independent colour specifiers to a rectangular array of pixels. nx and ny dimension the rectangular array of device independent colour specifiers. (Here and elsewhere in this discussion we are refering to the absolute value of nx and ny since the signs of these values (see below) are used to indicate the directions in which succeeding pixels are drawn.)

The origin is the location in VDC space of the drawing bitmap at which the first colour value is to be placed.
nx is a signed integer, the absolute value of which defines how many pixels are in each row. If nx>0, then the row extends toward increasing VDC X from the origin point. If nx<0, then the row extends toward decreasing VDC X from the origin point.

ny is a signed integer, the absolute value of which defines how many rows of pixels there are. If ny>0, then each new row is toward increasing VDC Y. If ny<0, then each new row is toward decreasing VDC Y.

valid x range and valid y range specify the subrectangle within the array which is to be drawn into the bitmap (refer to figure 6). This allows a rectangular subregion of the pixel array to be rendered. If the valid x or y range extends beyond the limits of the nx by ny array of colour specifiers, then the valid x or y range is "clipped" to the nx by ny limits of the array, with only those values in the valid x and y ranges which are also in the limits of the colour specifier array being rendered.

x scale and y scale permit integer scaling of the pixel array rendering independently in the x and y dimensions (see figure 6). These scale factors must be positive integers.

If nx or ny or x scale or y scale = 0, nothing is drawn.

Add the following sub-clauses after sub-clause 5.7.35

5.7.36 LINE REPRESENTATION

Parameters:

line bundle index (IX)
line type indicator (IX)
line width specifier, either
absolute line width (VDC)
or
line width scale factor (R)
line colour specifier, either
line colour index (CI)
or
line colour value (CD)

Description:

In the line bundle table, the given line bundle index is associated with the specific parameters.

Line type is specified and behaves as indicated in the LINE TYPE attribute function.

Line width is defined in the current LINE WIDTH SPECIFICATION MODE and is stored in the bundle table along with that mode. Thus, the definition is immune to subsequent changes to the selection mode.

The line bundle table has predefined entries. Each entry renders a distinct appearance from other predefined entries. Any table entry (including the predefined entries) may be redefined with this function. Redefining a table entry or adding a new table entry may eliminate the ability to render a distinct appearance from other table entries.

When line functions are displayed the line bundle index refers to an entry in the line bundle table.

Which aspects in the entry are used depends upon the setting of the corresponding ASFs, see the ASPECT SOURCE FLAGS function.

5.7.37 MARKER REPRESENTATION

Parameters:

marker bundle index (IX)
marker type indicator (IX)
marker size specifier, either
absolute marker size (VDC)
or
marker size scale factor (R)
marker colour specifier, either
marker colour index (CI)
or
marker colour value (CD)

Description:

In the marker bundle table, the given marker bundle index is associated with the specified parameters.

Marker type is specified and behaves as indicated in the MARKER TYPE attribute function.

Marker size is defined in the current MARKER SIZE SPECIFICATION MODE and is stored in the bundle table along with that mode. Thus, the definition is immune to subsequent changes to the specification mode.

Marker colour is defined in the current COLOUR SELECTION MODE, and is stored in the bundle table along with that mode. Thus the definition is immune to subsequent changes to the selection mode.

The marker bundle table has predefined entries. Each entry renders a distinct appearance from other predefined entries. Any table entry (including the predefined entries) may be redefined with this function. Redefining a table entry or adding a new table entry may eliminate the ability to render a distinct appearance from other table entries.

When polymarkers are displayed the marker bundle index refers to an entry in the marker bundle table.

Which aspects in the entry are used depends upon the setting of the corresponding ASFs, see the ASPECT SOURCE FLAGS function.

5.7.38 TEXT REPRESENTATION

Parameters:

text bundle index (IX)
text font index (IX)
text precision (one of: string, character, stroke) (E)
character expansion factor (R)
character spacing (R)
text colour specifier, either
    text colour index (CI)
        or
    text colour value (CD)

Description:

In the text bundle table, the given text bundle index is associated with the specified parameters.

Text font index is specified and behaves as indicated in the TEXT FONT INDEX attribute function.

Text precision is specified and behaves as indicated in the TEXT PRECISION attribute function.

Character expansion factor is specified and behaves as indicated in the CHARACTER EXPANSION FACTOR attribute function.

Character spacing is specified and behaves as indicated in the CHARACTER SPACING attribute function.

Text colour is defined in the current COLOUR SELECTION MODE, and is stored in the bundle table along with that mode. Thus, the definition is immune to subsequent changes to the selection mode.

The text bundle table has predefined entries. Each entry renders a distinct appearance from other predefined entries. Any table entry (including the predefined entries) may be redefined with this function. Redefining a table entry or adding a new table entry may eliminate the ability to render a distinct appearance from other table entries.

When text is displayed the text bundle index refers to an entry in the text bundle table.

Which aspects in the entry are used depends upon the setting of the corresponding ASFs, see the ASPECT SOURCE FLAGS function.

5.7.39 FILL REPRESENTATION

Parameters:

fill area bundle index (IX)
interior style (one of: hollow, solid, pattern,hatch, empty)(E)
fill colour specifier, either
fill colour index (CI)
or
fill colour value (CD)
hatch index (IX)
pattern index (IX)

Description:

In the fill bundle table, the given fill bundle index is associated with the specified parameters.

Interior style is specified and behaves as indicated in the INTERIOR STYLE attribute function.

Hatch index indicator is specified and behaves as indicated in the HATCH INDEX attribute function.

Pattern index indicator is specified and behaves as indicated in the PATTERN INDEX attribute function.

Fill colour is defined in the current COLOUR SELECTION MODE, and is stored in the bundle table along with that mode. Thus, the definition is immune to subsequent changes to the selection mode.

The fill bundle table has predefined entries. Each entry renders a distinct appearance from other predefined entries. Any table entry (including predefined entries) may be redefined with this function. Redefining a table entry or adding a new table entry may eliminate the ability to render a distinct appearance from other table entries.

When fill areas are displayed the fill bundle index refers to an entry in the fill bundle table.

Which aspects in the entry are used depends upon the setting of the corresponding ASFs, see the ASPECT SOURCE FLAGS function.

5.7.40 EDGE REPRESENTATION

Parameters:

edge bundle index (IX)
edge type indicator (IX)
edge width specifier, either
absolute edge width (VDC)
or
edge width scale factor (R)
edge colour specifier, either
edge colour index (CI)
or
edge colour value (CD)

Description:

In the edge bundle table, the given edge bundle index is associated with the specified parameters.

Edge type is specified and behaves as indicated in the EDGE TYPE attribute function.

Edge width is defined in the current EDGE WIDTH SPECIFICATION MODE and is stored in the bundle table along with that mode. Thus, the definition is immune to subsequent changes to the specification mode.

Edge colour is defined in the current COLOUR SELECTION MODE and is stored in the bundle table along with that mode. Thus, the definition is immune to subsequent changes to the selection mode.

The edge bundle table has predefined entries. Each entry renders a distinct appearance from other predefined entries. Any table entry (including predefined entries) may be redefined with this function. Redefining a table entry or adding a new table entry may eliminate the ability to render a distinct appearance from other table entries.

When fill areas are displayed the edge bundle index refers to an entry in the edge bundle table.

Which aspects in the entry are used depends upon the setting of the corresponding ASFs, see the ASPECT SOURCE FLAGS function.

5.7.41 PICK IDENTIFIER

Parameters:

pick identifier (I)

Description:

The pick identifier is associated with all of the graphical primitive elements of a segment until the next PICK IDENTIFIER element.

With pick input, a structure is returned consisting of the picked segment identifier and a pick identifier. This pick identifier represents the graphical elements that were associated with it during creation of the segment. This pick structure is returned only if the picked segment is both VISIBLE and DETECTABLE. The default pick identifier is zero.

5.7.42 DRAWING MODE

Parameters:

drawing mode (I)

Description:

Drawing mode is an integer in the range [0..15] which defines the logical operation between the source and destination during all output operations.

TABLE XX. Drawing modes

The 16 possible combinations are defined as follows
(d' = resulting destination bit value,
d = original destination bit value,
s = original source bit value):

Drawing Mode	Logical Operation
0	d' = 0
1	d' = s AND d
2	d' = s AND (NOT d)
3	d' = s
4	d' = (NOT s) AND d
5	d' = d
6	d' = s XOR d
7	d' = s OR d
8	d' = NOT (s OR d)
9	d' = NOT (s XOR d)
10	d' = NOT d
11	d' = s OR (NOT d)
12	d' = NOT s
13	d' = (NOT s) OR d
14	d' = NOT (s AND d)
15	d' = 1

Add the following sub-clause after sub-clause 5.9:

5.10 Segment Elements

5.10.1 REOPEN SEGMENT

Parameters:

segment identifier (SN)

Description:

The segment corresponding to the specified identifier is reopened. Subsequent graphical objects are appended to the segment.

Issuing a REOPEN SEGMENT with an identifier which is not currently in use is an error and the function is ignored.

The display of graphical objects which are added to an existing segment depends on the segment visibility, the segment highlighting, the implicit segment regeneration mode, and the segment display priority.

5.10.2 COPY SEGMENT

Parameters:

segment identifier	(SN)
copy transformation matrix:	
scaling and rotation portion	2x2xR
translation portion	2x1xVDC

Description:

All graphical objects in the identified segment are reentered into the pipeline and placed into the open segment. The identified segment is referred to as the copied segment. The segment attributes of the copied segment are ignored. The segment attributes of the open segment are unchanged by the COPY SEGMENT function.

The copy transformation is applied to all graphical objects of the copied segment before they are copied into the open segment. The copy transformation is not applied to clipping rectangles.

The COPY SEGMENT function duplicates all of the graphical objects of the copied segment as though the client had invoked CGI functions to define the objects as transformed.

The display of graphical objects which are copied into the open segment depends on the segment visibility, the segment highlighting, the implicit segment regeneration mode, and the segment display priority.

The INHERITANCE FILTER function allows for control of the attribute values which are used when copying segments. This filter controls whether individual attribute values are reapplied to the graphical objects.

5.10.3 DELETE SEGMENT

Parameters:

segment identifier (SN]

Description:

The identified segment is deleted.

NOTE - The segment identifier may appear in a subsequent BEGIN SEGMENT element.

5.10.4 DELETE ALL SEGMENTS

Parameters:

None

Description:

All segments are deleted from segment storage. All segment identifiers may be reused by the BEGIN SEGMENT.

Display changes resulting from the deletion of all segments are governed by the implicit segment regeneration mode and dynamic modification accepted for segment deletion.

If the description table entry "Dynamic modification accepted for Segment Deletion" is set to IMM, the effect of DELETE ALL SEGMENTS happens immediately. If the dynamic modification entry is set to IRG, the effect of DELETE ALL SEGMENTS depends on the setting of the "Implicit Segment Regeneration Mode" entry in the state list, as follows:

If the implicit segment regeneration mode is ALLOWED, any image modification necessary as a result of the DELETE ALL SEGMENTS function will be achieved by an immediate implicit regeneration.

If the implicit segment regeneration mode is SUPPRESSED, any image modification necessary as a result of the DELETE ALL SEGMENTS function will not take place until some explicit action is performed.

5.10.5 RENAME SEGMENT

Parameters:

old segment identifier (SN)
new segment identifier (SN)

Description:

An existing segment is associated with a new segment identifier.

5.10.6 REDRAW ALL SEGMENTS

Parameters:

None

Description:

This function is intended to result in a redraw of all defined segments. However, if a segment's visibility attribute is INVISIBLE, that segment is not drawn. Segments of higher display priority should always appear to cover overlapping segments of lower display priority.

5.10.7 IMPLICIT SEGMENT REGENERATION MODE

Parameters:

implicit segment regeneration mode (one of: SUPPRESSED, ALLOWED)(E)

Description:

The implicit segment regeneration mode for picture changes is set to SUPPRESSED or ALLOWED. The IMPLICIT SEGMENT REGENERATION MODE function may be issued at any time.

If the mode is ALLOWED then the interpreter may perform an implicit PREPARE VIEW SURFACE and REDRAW ALL SEGMENTS depending on the dynamic modification accepted values. In this

mode, an open segment (or any other segment) may be redrawn whenever changes affect the appearance of the view surface. In such cases, all non-retained data are lost.

When set to SUPPRESSED, the interpreter must explicitly invoke PREPARE VIEW SURFACE and REDRAW ALL SEGMENTS or REDRAW SEGMENT to redraw graphical objects stored in segments. If a segment is open, it must be closed (END SEGMENT) prior to invoking explicit segment display functions.

Some devices cannot immediately change a picture. A plotter for example can only add to a picture. It would need to advance the paper and redraw the picture to show the effect of a transformation change. Changes in display priority could cause different sections of a picture to become visible or obscured. Visibility and highlighting may also cause the picture to change.

On devices where pictures are additive only (i.e., cannot erase or change part of a picture) implicit segment regeneration mode of SUPPRESSED is more efficient in terms of time and material. Implicit segment regeneration mode gives the client the ability to accumulate picture changes. At some later time, the client can explicitly utilize PREPARE VIEW SURFACE and REDRAW ALL SEGMENTS.

The default IMPLICIT SEGMENT REGENERATION MODE is SUPPRESSED.

5.10.8 INHERITANCE FILTER

Parameters:

filter selection attribute designator (list of:
{individual function names},
(See below)
LINE ATTRIBUTES,
MARKER ATTRIBUTES,
TEXT ATTRIBUTES,
CHARACTER ATTRIBUTES
FILL ATTRIBUTES,
EDGE ATTRIBUTES,
PATTERN ATTRIBUTES,
CLIP CONTROL,
OUTPUT CONTROL,
ALL)(nE)
setting (STATE_LIST,SEGMENT)(E)

Description:

The setting of the inheritance filter is modified for those attributes in the filter selection list. According to the setting, attributes are inherited from the current state lists or from the copied segment.

The individual function names permitted are those listed in Table XX, "Inheritance Filter Function Groups", as one of the attributes included.

If an attribute or group of attributes designated in the filter selection list is set to STATE_LIST, graphical objects inherit that attribute or group of attributes from the current state lists when a segment is copied.

If an attribute or group of attributes designated in the filter selection list is set to SEGMENT, that attribute or group of attributes is unaffected (in all graphical objects employing them) by the corresponding CGI state list when a segment is copied.

The default inheritance filter setting value is SEGMENT for all attributes.

The groups of attributes are defined in the table below.

TABLE XX Inheritance Filter Function Groups

Attribute Group	Attributes Included

LINE ATTRIBUTES	LINE BUNDLE INDEX
	LINE TYPE
	LINE WIDTH
	LINE COLOUR
MARKER ATTRIBUTES	MARKER BUNDLE INDEX
	MARKER TYPE
	MARKER SIZE
	MARKER COLOUR
TEXT ATTRIBUTES	TEXT BUNDLE INDEX
	TEXT FONT INDEX
	TEXT PRECISION
	CHARACTER EXPANSION FACTOR
	CHARACTER SPACING
	TEXT COLOUR
CHARACTER ATTRIBUTES	CHARACTER HEIGHT
	CHARACTER ORIENTATION
	TEXT PATH
	TEXT ALIGNMENT
	CHARACTER SET INDEX
	ALTERNATE CHARACTER SET INDEX
FILL ATTRIBUTES	FILL BUNDLE INDEX
	INTERIOR STYLE
	FILL COLOUR
	HATCH INDEX
	PATTERN INDEX
EDGE ATTRIBUTES	EDGE BUNDLE INDEX
	EDGE TYPE
	EDGE WIDTH
	EDGE COLOUR
	EDGE VISIBILITY
PATTERN ATTRIBUTES	FILL REFERENCE POINT
	PATTERN SIZE
CLIP CONTROL	CLIP RECTANGLE
	CLIP INDICATOR
OUTPUT CONTROL	AUXILIARY COLOUR
	TRANSPARENCY
	DRAWING MODE
ALL	All attributes controlled

5.10.9 SEGMENT TRANSFORM

Parameters:

segment identifier (SN)
transformation matrix (4R 2VDC)

Description:

The matrix is stored in the identified segment as a segment attribute. The segment transform replaces the old segment transform. There is no accumulation of matrices.

When a segment is displayed, the segment transform is applied to all reference points in VDC space with the following matrix

$$\begin{vmatrix} X' \\ \\ Y' \end{vmatrix} = \begin{vmatrix} M11 & M12 & M13 \\ & & \\ M21 & M22 & M23 \end{vmatrix} \cdot \begin{vmatrix} X \\ Y \\ 1 \end{vmatrix}$$

where X an Y is the original coordinate pair and X' and Y' is the new coordinate pair.

Reference points may refer to both output primitive coordinate pairs as well as to geometric attributes. Note that the reference points for all output primitives are defined to be input

parameters of type POINT. In the case of reference points for geometric attributes that are of vectors, such as CHARACTER ORIENTATION, the translation portion of the matrix M13 and M23 is not applied.

The default segment transform is the identity matrix. The segment transform may be set after a segment has been open or created. It is permissible to change the transform of the open segment.

5.10.10 SEGMENT VISIBILITY

Parameters:

segment identifier	(SN)
visibility (one of: visible, invisible)	(E)

Description:

When the visibility attribute is set to 'visible', the segment may be displayed. When this attribute is set to 'invisible' the segment must not be displayed.

NOTE - Invisible segments cannot be picked.

5.10.11 SEGMENT HIGHLIGHTING

Parameters:

segment identifier (SN)
highlighting (one of: normal, highlighted) (E)

Description:

When the highlighting attribute is set to 'highlighted', the visual appearance of the segment is implementation dependent. When the highlighting attribute is set to 'normal', the segment is displayed according to the segment and primitive attributes.

5.10.12 SEGMENT DISPLAY PRIORITY

Parameters:

segment identifier	(SN)
segment display priority	(I)

Description:

The segment display priority for the identified segment is set to the specified value.

Segments with higher segment display priority appear to be in front of segments with lower segment display priorities. When the segment display priorities of two overlapping segments are the same, the order in which they appear is implementation dependent.

5.10.13 SEGMENT DETECTABILITY

Parameters:

segment identifier	(SN)
detectability (one of: detectable, undetectable)	(E)

Description:

When the detectability attribute is set to 'detectable' and the visibility attribute to 'visible', the segment can be picked. 'detectable' but 'invisible' or 'undetectable' segments cannot be picked.

5.10.14 SEGMENT PICK PRIORITY

Parameters:

segment identifier	(SN)
segment pick priority	(I)

Description:

The segment pick priority for the identified segment is set to the specified value.

This value is used to determine which segment is to be picked when segments overlap. The segment with the highest pick priority is picked first.

Page 103

Clause 6: Add the following at the end of clause 6:

METAFILE CATEGORY	basic cgm
MAXIMUM VDC EXTENT	0,32767 for VDC type integer 0.0,1.0 for VDC type real
SEGMENT PRIORITY EXTENT	0-?
DEVICE VIEWPORT	i.d.
DEVICE VIEWPORT SPECIFICATION MODE	fraction of display surface
DEVICE VIEWPORT MAPPING	forced,left,bottom
DEFERRAL MODE	asap
UPDATE	
IMPLICIT EDGE VISIBILITY	off
LINE REPRESENTATION	i.d.
MARKER REP	i.d.
TEXT REPRESENTATION	i.d.
FILL REPRESENTATION	i.d.
EDGE REPRESENTATION	i.d.
PICK IDENTIFIER	n/a
DRAWING MODE	3 (d'=s)
IMPLICIT SEGMENT REGENERATION MODE	suppressed
INHERITANCE FILTER	segment
SEGMENT TRANSFORM	1,0,0 0,1,0
SEGMENT VISIBILITY	visible
SEGMENT HIGHLIGHTING	normal
SEGMENT DISPLAY PRIORITY	0

SEGMENT DETECTABILITY	undetectable
SEGMENT PICK PRIORITY	0

Page 104

Add the following clause after clause 7.4

7.5 Conformance by Metafile Category

A metafile must conform to the metafile category defined in the Metafile Descriptor. This conformance is to the abstract specification and encodings in the way described above. Conformance to a metafile category occurs when the metafile also conforms to the formal grammar for that category.

Page 127

Add the following text to the end of clause D4.4

SCALING MODE and DEVICE VIEWPORT

If both device viewport and scaling mode appear in the same metafile then the last specified is used. If neither appear then the default values for device viewport take precedence where these are allowed in the same category.

Page 132

Add the following clause after clause D.4.8

D.4.9 Segment Elements

If there is an attempt to open a segment which exists then the OPEN is ignored.

Page 144

Add the following clauses after sub-clause E.6

E.7 GKS Item Types

<This will be the list from Annex E of GKS>

E.8 The Relationship of Fonts Between CGM and GKS

The GKS standard includes the concepts of text output primitive attributes. However, the mechanism for specifying the text font differs from that specified in the CGM standard. This clause suggests an approach to handling these attributes within the GKS environment when the CGM is used as a GKS Metafile.

E.8.1 Overview of the Differences Between GKS and CGM Fonts

While CGM supports the TEXT output primitive attribute functionaility of GKS, a one-to-one mapping between CGM and GKS is not possible in all cases. Specifically:

1. GKS and CGM differ in the way fonts are defined:

   In the CGM text fonts are defined with the FONT ILIST element that associates font names or identifications with entries in a Font Table. The Font Table can be modified with the MODIFY FONT LIST element.

   In GKS, no mechanism is available for defining text fonts. GKS associates a unique text font number with each font. The Registration Authority is responsible for defining this mapping of font numbers to specific font identifications.

2. GKS and CGM differ in the way fonts are selected

In the CGM, text fonts are selected with the TEXT FONT INDEX element. The index selects an indeividual font from different fonts in the font list.

In GKS, text fonts are selected with a font number. The font number selects a specific GKS registered font if the value is positive. If the font number is negative an implementation dependent font is selected.

3. GKS and CGM differ on the independence of font and text precision

In the CGM, the font and text precision are specified by independent elements.

In GKS, the font and text precision are directly associated with specification by a single function.

4. GKS and CGM differ in character orientation capability

In CGM, the character orientation is specified by both a Character Up Vector and a Character Base Vector. Skewed specifications are possible.

In GKS, the character orientation is specified by a Character Up Vector. The Character Base Vector is always equal to positive 90 degrees from the Character Up Vector. Skewed specification is not possible.

5. Some CGM Elements have no counterpart as GKS functions

These include: character set related elements, such as CHARACTER SET LIST, CHARACTER CODING ANNOUNCER, CHARACTER SET INDEX, ALTERNATE CHARACTER SET INDEX; and Auxiliary Colour related elements, such as AUXILLIARY COLOUR and TRANSPARENCY that affect the presentation of text.

This additional functionality of the CGM causes no special problems for a GKS environment interpreting a CGM of category GKSM since the category restricts the occurrence of the elements associated with this additional functionality. Thus, the interpretation by GKS of a CGM of categorm GKSM is well defined.

E.8.2 Suggestion for Interpretation of CGM Font Information by GKS

GKS environments interpreting a CGM specify fonts with a font number. It is assumed that GKS maintains a list associating positive font numbers with a GKS registered font name or identifier. Private font numbers (i.e. ngative values) must be maintained in an implementation dependent list of associations. As the FONT LIST element is interpreted, an additaional list must be maintained that associates indeividualfont names specified in the CGM with a font index. The subsequent interpretation of the MODIFY FONT LIST element will result in changes or additions to this list. When the TEXT FONT INDEX element is interpretted, the font name associated with the font index is determined from the list of currently used fonts. the font name is used to determine the GKS font number associated with this font from a list of GKS registered fonts. This font number is used as the font parameter of the TEXT FONT AND PRECISION function. The value of the precision parameter is taken from the TEXT PRECISION element that directly follows each TEXT FONT INDEX element in a CGM with category GKSM.

E.9 Character Vectors

A metafile of GKSM category writes the CHARACTER HEIGHT and CHARACTER ORIENTATION as a pair in the CGM in that order. One interpretation this will return the character vector information back to the GKS application using the following mapping:

(CH,CO) maps to $CH \cdot \frac{\text{Character Up Vector}}{|\text{Character Up Vector}|}$ $CH \cdot \frac{\text{Character Base Vector}}{|\text{Character Up Vector}|}$

The following annex forms a new annex F.

F Formal Grammar of the Functional Specification of the CGMEXT1 Category

NOTE - This annex is not part of the Standard; it is included for information purposes only.

F.1 Introduction

This grammar is a formal definition of a standard CGM extended syntax. The encoding-independent and the encoding-dependent productions are separated, and there are subsections showing the syntax of each of the standardized encoding schemes. Details on the encoding of terminal symbols can be found in parts of this Standard that deal with the particular encoding schemes.

F.2 Notation Used

<symbol>	- nonterminal
<SYMBOL>	- terminal
<symbol>*	- 0 or more occurrences
<symbol>+	- 1 or more occurrences
<symbol>o	- optional (0 or 1 occurrences)
<symbol>(n)	- exactly n occurrences, n=2,3,...
<symbol-1> ::= <symbol-2>	- symbol-1 has the syntax of symbol-2
<symbol-1> | <symbol-2>	- symbol-1 or alternatively symbol-2
<symbol: meaning>	- symbol with the stated meaning
{comment}	- explanation of a symbol or a production

F.3 Detailed Grammar

F.3.1 Metafile Structure

```
<metafile>              ::= <BEGIN METAFILE>
                                <identifier>
                            <metafile descriptor>
                            <metafile contents>
                            <END METAFILE>

<metafile contents>     ::= <extra element>*
                            <picture>
                            <extra element>*

<extra element>         ::= <external element>
                          | <escape element>

<picture>               ::= <BEGIN PICTURE>
                                <identifier>
                            <picture descriptor element>*
                            <BEGIN PICTURE BODY>
                            <picture content>*
                            <END PICTURE>

<picture content>       ::= <picture element>
                          | <segment>

<identifier>            ::= <string>

<picture element>       ::= <control element>
                          | <graphical element>
                          | <attribute element>
```

```
                                  | <escape element>
                                  | <external element>
                                  | <segment element>

<segment>                         ::= <BEGIN SEGMENT>
                                          <segment name>
                                      <eligible picture element>*
                                      <END SEGMENT>

<segment name>                    ::= <integer>

<eligible picture element>        ::= <eligible local segment picture element>
                                    | <eligible global segment picture element>

<eligible local seg picture elem> ::= <expand from state table>
<eligible global seg picture elem> ::= <expand from state table>
```

F.3.2 Metafile Descriptor Elements

```
<metafile descriptor>             ::= <identification>
                                      <characteristics>

<identification>                  ::= <METAFILE VERSION>
                                          <integer>
                                      <metafile description>o
                                          <metafile category>o

<metafile category>               ::= <METAFILE CATEGORY>
                                          <category enumerated>

<metafile description>            ::= <METAFILE DESCRIPTION>
                                          <string>

<category enumerated>             ::= <CGM>
                                    | <CGMEXT1>
                                    | <GKSM>

<characteristics>                 ::= <element list>
                                      <optional descr elmt>*

<element list>                    ::= <METAFILE ELEMENT LIST>
                                          <element name>*

<optional descr elmt>             ::= <VDC TYPE>
                                          <vdc type>
                                    | <MAXIMUM COLOUR INDEX>
                                          <colour index>
                                    | <COLOUR VALUE EXTENT>
                                          <red green blue>(2)
                                    | <METAFILE DEFAULTS REPLACEMENT>
                                          <element default>+
                                    | <FONT LIST>
                                          <font name>+
                                    | <CHARACTER SET LIST>
                                          <character set definition>+
                                    | <CHARACTER CODING ANNOUNCER>
                                          <coding technique enumerated>
                                    | <scalar precision>*
                                    | <escape element>
                                    | <external element>
                                    | <MAXIMUM VDC EXTENT>
                                          <point> (2)
                                    | <SEGMENT PRIORITY EXTENT>
                                          <minimum extent>
```

```
                                        <maximum extent>

<vdc type>                        ::=  <INTEGER>
                                    |  <REAL>

<element default>                 ::=  <eligible control element>
                                    |  <picture descriptor element>
                                    |  <attribute element>
                                    |  <escape element>

<font name>                       ::=  <string>

<character set definition>        ::=  <char set enumerated>
                                            <designation sequence>

<index>                           ::=  <standard index value>
                                    |  <private index value>

<standard index value>            ::=  <non-negative integer>
<non-negative integer>            ::=  <integer>    {greater or equal to 0}
<positive integer>                ::=  <integer>    {greater than 0}
<private index value>             ::=  <negative integer>
<negative integer>                ::=  <integer>    {less than 0}
<positive index value>            ::=  <positive integer>

<char set enumerated>             ::=  <94 CHAR>
                                    |  <96 CHAR>
                                    |  <MULTI-BYTE 94 CHAR>
                                    |  <MULTI-BYTE 96 CHAR>
                                    |  <COMPLETE CODE>

<coding technique enumerated>     ::=  <BASIC 7-BIT>
                                    |  <BASIC 8-BIT>
                                    |  <EXTENDED 7-BIT>
                                    |  <EXTENDED 8-BIT>

<designation sequence>            ::=  <string>

<scalar precision>                ::=  <INTEGER PRECISION>
                                            <integer precision value>
                                    |  <REAL PRECISION>
                                            <real precision value>
                                    |  <INDEX PRECISION>
                                            <index precision value>
                                    |  <COLOUR PRECISION>
                                            <colour precision value>
                                    |  <COLOUR INDEX PRECISION>
                                            <col index precision value>
                                       {these elements have encoding}
                                       {dependent parameters      }

<eligible control element>        ::=  <VDC PRECISION>
                                    |  <AUXILIARY COLOUR>
                                    |  <TRANSPARENCY>
                                    |  <CLIP RECTANGLE>
                                    |  <CLIP INDICATOR>
                                    |  <VDC EXTENT>
                                    |  <DEVICE VIEWPORT>
                                    |  <DEVICE VIEWPORT SPECIFICATION MODE
                                    |  <DEVICE VIEWPORT MAPPING>
                                    |  <SET DEFERRAL MODE>
                                    |  <UPDATE>
                                    |  <IMPLICIT EDGE VISIBILITY>
```

```
<point>                          ::= <vdc value> (2)

<minimum extent                  ::= <integer>

<maximum extent>                 ::= <integer>
```

F.3.3 Picture Descriptor Elements

```
<picture descriptor element>     ::= <SCALING MODE>
                                         <scaling spec mode>
                                         <metric scale factor>
                                   | <COLOUR SELECTION MODE>
                                         <colour select mode>
                                   | LINE WIDTH SPECIFICATION MODE>
                                         <spec mode>
                                   | <MARKER SIZE SPECIFICATION MODE>
                                         <spec mode>
                                   | <EDGE WIDTH SPECIFICATION MODE>
                                         <spec mode>
                                   | <VDC EXTENT>
                                         <point>(2)
                                   | <BACKGROUND COLOUR>
                                         <red green blue>
                                   | <escape element>
                                   | <external element>

<colour select mode>             ::= <INDEXED>
                                   | <DIRECT>

<scaling spec mode>              ::= <ABSTRACT>
                                   | <METRIC>

<metric scale factor>            ::= <real>

<spec mode>                      ::= <ABSOLUTE>
                                   | <SCALED>

<point>                          ::= <vdc value> (2)
```

F.3.4 Control Elements

```
<control element>                ::= <vdc precision>
                                   | <AUXILIARY COLOUR>
                                         <colour>
                                   | <TRANSPARENCY>
                                         <on-off indicator enumerated>
                                   | <CLIP RECTANGLE>
                                         <point>(2)
                                   | <CLIP INDICATOR>
                                         <on-off indicator enumerated>
                                   | <VDC EXTENT>
                                         <point>(2)
                                   | <DEVICE VIEWPORT>
                                         <device point>(2)
                                   | <DEVICE VIEWPORT SPECIFICATION MODE
                                         <VSU specifier>
                                         <metric scale factor>
                                   | <DEVICE VIEWPORT MAPPING>
                                         <isotropy flag>
                                         <horizontal alignment flag>
                                         <vertical alignment flag>
                                   | <SET DEFERRAL MODE>
                                         <deferral mode enumerated>
```

```
                                  | <MAKE PICTURE CURRENT>
                                  | <PREPARE VIEW SURFACE>
                                      <force hard copy advance>
                                  | <UPDATE>
                                      <update regeneration flag enumerated>
                                  | <MODIFY FONT LIST>
                                      <starting index>
                                      <font name list>
                                  | <MODIFY CHARACTER SET LIST>
                                      <starting index>
                                      <character set definition list>
                                  | <BEGIN FIGURE>
                                  | <END FIGURE>
                                  | <NEW REGION>
                                  | <IMPLICIT EDGE VISIBILITY>
                                      <implicit edge visibility>
                                  | <SAVE PRIMITIVE ATTRIBUTES>
                                      <attribute set name>
                                  | <RESTORE PRIMITIVE ATTRIBUTES>
                                      <attribute set name>
                                  | <DELETE ATTRIBUTE SET>
                                      <attribute set name>

<on-off indicator enumerated>     ::= <ON>
                                    | <OFF>

<colour>                          ::= <colour index>
                                    | <red green blue>

<vdc precision>                   ::= <VDC INTEGER PRECISION>
                                          <vdc integer precision value>
                                    | <VDC REAL PRECISION>
                                          <vdc real precision value>
                                      {these elements have encoding}
                                      {dependent parameters       }

<device point>                    ::=<real>(2)

<VSU specifier enumerated>        ::= <FRACTION OF DEFAULT DEVICE VIEWPORT>
                                    | <MILLIMETRES WITH SCALE FACTOR>
                                    | <PHYSICAL DEVICE UNITS>

<metric scale factor>               | <real>

<isotropy flag enumerated>        ::= <NOT FORCED>
                                    | <FORCED>

<horiz align flag enumer>         ::= <LEFT>
                                    | <CENTRE>
                                    | <RIGHT>

<vert align flag enumer>          ::= <BOTTOM>
                                    | <CENTRE>
                                    | <TOP>

<deferral mode enumerated>        ::= <ASAP>
                                    | <BNI>
                                    | <ASTI>

<force hard copy advance
    enumerated>                   ::= <FORCE HARDCOPY>
                                      <CONDITIONAL>
```

```
<update regeneration flag
   enumerated>          ::= <PERFORM>
                          | <POSTPONE>

<starting index>        ::= <index>

<font name list>        ::= <font name>+

<character set def. list>  ::= <character set definition>+

<implicit edge vis enumer>  ::= <OFF>
                          | <ON>

<attribute set name>    ::= <string>
```

F.3.5 Graphical Elements

```
<graphical element>     ::= <polypoint element>
                          | <text element>
                          | <cell element>
                          | <gdp element>
                          | <rectangle element>
                          | <circular element>
                          | <elliptical element>
                          | <pixel array element>

<polypoint element>     ::= <POLYLINE>
                                <point pair>
                                <point list>
                          | <DISJOINT POLYLINE>
                                <point pair>
                                <point pair list>
                          | <POLYMARKER>
                                <point>
                                <point list>
                          | <POLYGON>
                                <point>(3)
                                <point list>
                          | <POLYGON SET>
                                <point edge pair>(3)
                                <point edge pair list>

<point list>            ::= <point>*

<point pair list>       ::= <point pair>*

<point pair>            ::= <point>(2)

<point edge pair>       ::= <point><edge out flag>

<point edge pair list>  ::= <point edge pair>*

<edge out flag>         ::= <INVISIBLE>
                          | <VISIBLE>
                          | <CLOSE_INVISIBLE>
                          | <CLOSE_VISIBLE>

<text element>          ::= <TEXT>
                                <point>
                                <text tail>
                          | <restricted text element>

<restricted text element>  ::= <RESTRICTED TEXT>
```

```
                                    <extent>
                                    <point>
                                    <text tail>

<extent>                      ::=   <vdc value>(2)

<text tail>                   ::=   <final character list>
                                  | <nonfinal character list>

<final character list>        ::=   <FINAL>
                                        <character list>

<nonfinal character list>     ::=   <NOT FINAL>
                                        <string>
                                        <character attribute element>*
                                        <spanned text>

<spanned text>                ::=   <APPEND TEXT>
                                        <text tail>

<cell element>                ::=   <CELL ARRAY>
                                        <point>(3)
                                        <integer>(2)
                                        <local colour precision>
                                        <colour>(integer1 x integer2)

                                    {this element has an encoding}
                                    {dependent parameter        }

<local colour precision>      ::=   <colour precision value>
                                  | <col index precision value>
                                  | <default col precision indicator>

<gdp element>                 ::=   <GDP>
                                        <gdp identifier>
                                        <point list>*
                                        <data record>

<gdp identifier>              ::=   <integer>

<rectangle element>           ::=   <RECTANGLE>
                                        <point pair>

<circular element>            ::=   <CIRCLE>
                                        <point>
                                        <radius>
                                  | <CIRCULAR ARC 3 POINT>
                                        <point>(3)
                                  | <CIRCULAR ARC 3 POINT CLOSE>
                                        <point>(3)
                                        <close type>
                                  | <CIRCULAR ARC CENTRE>
                                        <point>
                                        <vdc value>(4)
                                        <radius>
                                  | <CIRCULAR ARC CENTRE CLOSE>
                                        <close type>
                                  | <CIRCULAR ARC CENTRE BACKWARDS>
                                        <point>
                                        <vdc value>(4)
                                        <radius>

<radius>                      ::=   <non-negative vdc value>
```

```
<non-negative vdc value>      ::= <vdc value> {greater or equal to 0}

<close type>                  ::= <PIE>
                                | <CHORD>

<elliptical element>          ::= <ELLIPSE>
                                      <point>(3)
                                | <ELLIPTICAL ARC>
                                      <point>(3)
                                      <vdc value>(4)
                                | <ELLIPTICAL ARC CLOSE>
                                      <point>(3)
                                      <vdc value>(4)
                                      <close type>

<pixel array element>         ::= <PIXEL ARRAY>
                                      <point>
                                      <array dimension>
                                      <valid x range>
                                      <valid y range>
                                      <x scale>
                                      <y scale>
                                      <colour specifiers>

<array dimension>             ::= <integer>(2)

<valid x range>               ::= <integer>(2)

<valid y range>               ::= <integer>(2)

<x scale>                     ::= <integer>

<y scale>                     ::= <integer>

<colour specifiers>           ::= <array>
```

F.3.6 Attribute Elements

```
<primitive attribute elem>    ::= <line attribute element>
                                | <marker attribute element>
                                | <text attribute element>
                                | <filled area attribute element>
                                | <colour table element>
                                | <aspect source flags>
                                | <representation element>
                                | <pick identifier>
                                | <drawing mode>

<line attribute element>      ::= <LINE BUNDLE INDEX>
                                      <positive index>
                                | <LINE TYPE>
                                      <index>
                                | <LINE WIDTH>
                                      <size value>
                                | <LINE COLOUR>
                                      <colour>

<size value>                  ::= <non-negative vdc value>
                                | <non-negative real>

<non-negative real>           ::= <real> {greater or equal to 0}

<marker attribute element>    ::= <MARKER BUNDLE INDEX>
```

```
                                    <positive index>
                                  | <MARKER TYPE>
                                      <index>
                                  | <MARKER SIZE>
                                    <size value>
                                  | <MARKER COLOUR>
                                    <colour>

<text attribute element>        ::= <char attribute element>
                                  | <string attribute element>

<char attribute element>        ::= <TEXT BUNDLE INDEX>
                                      <positive index>
                                  | <TEXT FONT INDEX>
                                      <positive index>
                                  | <CHARACTER EXPANSION FACTOR>
                                      <real>
                                  | <CHARACTER SPACING>
                                      <real>
                                  | <TEXT COLOUR>
                                      <colour>
                                  | <CHARACTER HEIGHT>
                                      <non-negative vdc value>
                                  | <CHARACTER ORIENTATION>
                                      <vdc value>(4)
                                  | <CHARACTER SET INDEX>
                                      <positive index>
                                  | <ALTERNATE CHARACTER SET INDEX>
                                      <positive index>

<string attribute element>      ::= <TEXT PATH>
                                      <path enumerated>
                                  | <TEXT PRECISION>
                                      <text precision enumerated>
                                  | <TEXT ALIGNMENT>
                                      <horizontal align enumerated>
                                      <vertical align enumerated>
                                      <continuous align value> (2)

<path enumerated>               ::= <RIGHT>
                                  | <LEFT>
                                  | <UP>
                                  | <DOWN>

<text precision enumerated>     ::= <STRING>
                                  | <CHARACTER>
                                  | <STROKE>

<horizontal align enumerated>   ::= <NORMAL HORIZONTAL>
                                  | <LEFT>
                                  | <CENTRE>
                                  | <RIGHT>
                                  | <CONTINUOUS HORIZONTAL>

<vertical align enumerated>     ::= <NORMAL VERTICAL>
                                  | <TOP>
                                  | <CAP>
                                  | <HALF>
                                  | <BASE>
                                  | <BOTTOM>
                                  | <CONTINUOUS VERTICAL>

<continuous align value>        ::= <real>
```

```
<filled area attribute elem>      ::= <FILL BUNDLE INDEX>
                                          <positive index>
                                    | <INTERIOR STYLE>
                                          <interior style enumerated>
                                    | <FILL COLOUR>
                                          <colour>
                                    | <HATCH INDEX>
                                          <index>
                                    | <PATTERN INDEX>
                                          <positive index>
                                    | <EDGE BUNDLE INDEX>
                                          <positive index>
                                    | <EDGE TYPE>
                                          <index>
                                    | <EDGE WIDTH>
                                          <size value>
                                    | <EDGE COLOUR>
                                          <colour>
                                    | <EDGE VISIBILITY>
                                          <on-off indicator enumerated>
                                    | <FILL REFERENCE POINT>
                                          <point>
                                    | <PATTERN TABLE>
                                          <positive index>
                                          <integer>(2)
                                          <local colour precision>
                                          <colour>(integer1 x integer2)
                                      {this element has an encoding}
                                      {dependent parameter        }
                                    | <PATTERN SIZE>
                                          <vdc value>(4)

<interior style enumerated>       ::= <HOLLOW>
                                    | <SOLID>
                                    | <PATTERN>
                                    | <HATCH>
                                    | <EMPTY>

<colour table element>            ::= <COLOUR TABLE>
                                          <starting index>
                                          <red green blue>+

<starting index>                  ::= <colour index>

<aspect source flags>             ::= <ASPECT SOURCE FLAGS>
                                          <asf pair>+

<asf pair>                        ::= <asf type>
                                      <asf>

<asf type>                        ::= <LINE TYPE ASF>
                                    | <LINE WIDTH ASF>
                                    | <LINE COLOUR ASF>
                                    | <MARKER TYPE ASF>
                                    | <MARKER SIZE ASF>
                                    | <MARKER COLOUR ASF>
                                    | <TEXT FONT ASF>
                                    | <TEXT PRECISION ASF>
                                    | <TEXT FONT AND PRECISION ASF>
                                    | <CHARACTER EXPANSION FACTOR ASF>
                                    | <CHARACTER SPACING ASF>
                                    | <TEXT COLOUR ASF>
                                    | <INTERIOR STYLE ASF>
                                    | <FILL COLOUR ASF>
```

```
                                | <HATCH INDEX ASF>
                                | <PATTERN INDEX ASF>
                                | <EDGE TYPE ASF>
                                | <EDGE WIDTH ASF>
                                | <EDGE COLOUR ASF>

<asf>                       ::= <INDIVIDUAL>
                                | <BUNDLED>

<representation element>    ::= <LINE REPRESENTATION>
                                    <positive index>
                                    <index>
                                    <size value>
                                    <colour>
                                | <MARKER REPRESENTATION>
                                    <positive index>
                                    <index>
                                    <size value>
                                    <colour>
                                | <TEXT REPRESENTATION>
                                    <positive index>
                                    <index> {font}
                                    <text precision enumerated>
                                    <real>  {character spacing}
                                    <real>  {expansion factor}
                                    <colour>
                                | <FILL REPRESENTATION>
                                    <positive index>
                                    <interior style enumerated>
                                    <index>         {hatch index}
                                    <positive index>  {pattern index}
                                    <colour>
                                | <EDGE REPRESENTATION>
                                    <positive index>
                                    <index>
                                    <size value>
                                    <colour>
                                | <PATTERN TABLE>
                                    <positive index>
                                    <integer>(2)
                                    <local colour precision>
                                    <colour>(integer1*integer2)
                                  {this element has an encoding}
                                  {dependent parameter}
                                | <COLOUR TABLE>
                                    <starting index>
                                    <red green blue>+

<pick identifier>           ::= <PICK IDENTIFIER>
                                    <integer>

<drawing mode>              ::= <DRAWING MODE>
                                    <integer>         {0-15}
```

F.3.7 Escape Elements

```
<escape element>            ::= <ESCAPE>
                                    <identifier>
                                    <data record>

<identifier>                ::= <integer>
```

F.3.8 External Elements

```
<external element>          ::= <MESSAGE>
                                    <action flag>
                                    <string>
                              | <APPLICATION DATA>
                                    <integer>
                                    <data record>

<action flag>               ::= <YES>
                              | <NO>
```

F.3.9 Segment Elements

```
<segment element>           ::= <REOPEN SEGMENT>
                                    <segment name>
                              | <COPY SEGMENT>
                                    <segment name>
                                    <copy transformation matrix>
                              | <DELETE SEGMENT>
                                    <segment name>
                              | <DELETE ALL SEGMENTS>
                              | <RENAME SEGMENT>
                                    <old segment name>
                                    <new segment name>
                              | <REDRAW ALL SEGMENTS>
                              | <IMPLICIT SEGMENT REGENERATION MODE>
                                    <implicit segment regeneration mode enumerated>
                              | <INHERITANCE FILTER>
                                    <filter selection attrubute deisignator>+
                                    <setting>
                              | <SEGMENT TRANSFORM>
                                    <segment name>
                                    <transformation matrix>
                              | <SEGMENT VISIBILITY>
                                    <segment name>
                                    <visibility enumerated>
                              | <SEGMENT HIGHLIGHTING>
                                    <segment name>
                                    <highlighting enumerated>
                              | <SEGMENT DISPLAY PRIORITY>
                                    <segment name>
                                    <segment display priority>
                              | <SEGMENT DETECTABILITY>
                                    <segment name>
                                    <detectability enumerated>
                              | <SEGMENT PICK PRIORITY>
                                    <segment name>
                                    <segment pick priority>

<copy transformation matrix>  ::= <transformation matrix>

<transformation matrix>       ::= <real> (4)
                                  <vdc> (2)

<old segment name>            ::= <integer>

<new segment name>            ::= <integer>

<implicit seg regen mode enum>  ::= <SUPPRESSED>
                                  | <ALLOWED>
```

```
<filter selection att des enum>    ::= <LINE ATTRIBUTE>
                                     | <MARKER ATTRIBUTES>
                                     | <TEXT ATTRIBUTES>
                                     | <CHARACTER ATTRIBUTES>
                                     | <FILL ATTRIBUTES>
                                     | <EDGE ATTRIBUTES>
                                     | <PATTERN ATTRIBUTES>
                                     | <PATTERN ATTRIBUTES>
                                     | <CLIP CONTROL>
                                     | <OUTPUT CONTROL>
                                     | <ALL>

<setting enumerated>               ::= <STATE_LIST>
                                     | <SEGMENT>

<transformation matrix>            ::= <real>(6)

<visibility enumerated>            ::= <VISIBLE>
                                     | <INVISIBLE>

<highlighting enumerated>          ::= <NORMAL>
                                     | <HIGHLIGHTED>

<segment dispalay priority>        ::=<integer>

detectability enumerated           ::= <DETECTABLE>
                                     | <UNDETECTABLE>

<segment pick priority>            ::= <integer>
```

F.4 Terminal Symbols

The following are the terminals in this grammar.
Their representation is dependent on the encoding scheme used.
In annex A of the subsequent parts of this Standard, these encoding-dependent symbols are further described.

```
<element name>
<integer>
<real>
<vdc value>
<string>
<colour index>
<red green blue>
<integer prec value>
<real prec value>
<index prec value>
<colour prec value>
<col index prec value>
<default col prec indicator>
<vdc integer prec value>
<vdc real prec value>
<colour list>
<data record>
<device point>
<name>
<segment name>
<sttribute set name>
<viewport specification point>
```

The CGM extended opcodes are encoding dependent. A complete list of them can be found in the productions for <element name enumerated> below.

The enumerated types:

<CGM>
<CGMEXT1>
<GKSM>
<INTEGER>
<REAL>
<ON>
<OFF>
<INDEXED>
<DIRECT>
<ABSTRACT>
<METRIC>
<ABSOLUTE>
<SCALED>
<94 CHAR>
<96 CHAR>
<MULTI-BYTE 94 CHAR>
<MULTI-BYTE 96 CHAR>
<COMPLETE CODE>
<BASIC 7-BIT>
<BASIC 8-BIT>
<EXTENDED 7-BIT>
<EXTENDED 8-BIT>
<FRACTION OF DEFAULT DEVICE VIEWPORT>
<MILLIMETRES WITH SCALE FACTOR>
<PHYSICAL DEVICE UNITS>
<NOT FORCED>
<FORCED>
<LEFT>
<RIGHT>
<CENTRE>
<BOTTOM>
<TOP>
<ASAP>
<BNI>
<ASTI>
<FORCE HARDCOPY>
<CONDITIONAL>
<SUPPRESSED>
<ALLOWED>
<POSTPONE>
<PERFORM>
<CONDITIONALLY>
<ALWAYS>
<INVISIBLE>
<VISIBLE>
<CLOSE_INVISIBLE>
<CLOSE_VISIBLE>
<PIE>
<CHORD>
<FINAL>
<NOT FINAL>
<INDIVIDUAL>
<BUNDLED>
<HOLLOW>
<SOLID>
<PATTERN>
<HATCH>
<EMPTY>
<STRING>
<CHARACTER>
<STROKE>

<RIGHT>
<LEFT>
<UP>
<DOWN>
<NORMAL HORIZONTAL>
<CENTRE>
<CONTINUOUS HORIZONTAL>
<NORMAL VERTICAL>
<TOP>
<CAP>
<HALF>
<BASE>
<BOTTOM>
<CONTINUOUS VERTICAL>
<YES>
<NO>
<LINE TYPE ASF>
<LINE WIDTH ASF>
<LINE COLOUR ASF>
<MARKER TYPE ASF>
<MARKER SIZE ASF>
<MARKER COLOUR ASF>
<TEXT FONT ASF>
<TEXT PRECISION ASF>
<TEXT FONT AND PRECISION ASF>
<CHARACTER EXPANSION FACTOR ASF>
<CHARACTER SPACING ASF>
<TEXT COLOUR ASF>
<INTERIOR STYLE ASF>
<HATCH INDEX ASF>
<PATTERN INDEX ASF>
<FILL COLOUR ASF>
<EDGE TYPE ASF>
<EDGE WIDTH ASF>
<EDGE COLOUR ASF>
<LINE ATTRIBUTE>
<MARKER ATTRIBUTES>
<TEXT ATTRIBUTES>
<CHARACTER ATTRIBUTES>
<FILL ATTRIBUTES>
<EDGE ATTRIBUTES>
<PATTERN ATTRIBUTES>
<PATTERN ATTRIBUTES>
<CLIP CONTROL>
<OUTPUT CONTROL>
<ALL>
<STATE_LIST

<VISIBLE>
<INVISIBLE>
<NORMAL>
<HIGHLIGHTED>
<DETECTABLE>
<UNDETECTABLE>

<element name enumerated> ::= <BEGIN METAFILE>
| <END METAFILE>
| <BEGIN PICTURE>
| <BEGIN PICTURE BODY>
| <END PICTURE>
| <BEGIN SEGMENT>
| <END SEGMENT>
| <METAFILE VERSION>

<METAFILE CATEGORY>
<METAFILE DESCRIPTION>
<VDC TYPE>
<INTEGER PRECISION>
<REAL PRECISION>
<INDEX PRECISION>
<COLOUR PRECISION>
<COLOUR INDEX PRECISION>
<MAXIMUM COLOUR INDEX>
<METAFILE ELEMENT LIST>
<METAFILE DEFAULTS REPLACEMENT>
<FONT LIST>
<CHARACTER SET LIST>
<CHARACTER CODING ANNOUNCER>
<MAXIMUM VDC EXTENT>
<SEGMENT PRIORITY EXTENT>
<SCALING MODE>
<COLOUR SELECTION MODE>
<LINE WIDTH SPECIFICATION MODE>
<MARKER SIZE SPECIFICATION MODE>
<EDGE WIDTH SPECIFICATION MODE>
<VDC EXTENT>
<BACKGROUND COLOUR>
<VDC INTEGER PRECISION>
<VDC REAL PRECISION>
<AUXILIARY COLOUR>
<TRANSPARENCY>
<CLIP RECTANGLE>
<CLIP INDICATOR>
<DEVICE VIEWPORT>
<DEVICE VIEWPORT SPECIFICATION MODE
<DEVICE VIEWPORT MAPPING>
<SET DEFERRAL MODE>
<UPDATE>
<MODIFY FONT LIST>
<MODIFY CHARACTER SET LIST>
<BEGIN FIGURE>
<END FIGURE>
<NEW REGION>
<IMPLICIT EDGE VISIBILITY>
<SAVE PRIMITIVE ATTRIBUTES>
<RESTORE PRIMITIVE ATTRIBUTES>
<DELETE ATTRIBUTE SET>
<POLYLINE>
<DISJOINT POLYLINE>
<POLYMARKER>
<TEXT>
<RESTRICTED TEXT>
<APPEND TEXT>
<POLYGON>
<POLYGON SET>
<CELL ARRAY>
<GDP>
<RECTANGLE>
<CIRCLE>
<CIRCULAR ARC 3 POINT>
<CIRCULAR ARC 3 POINT CLOSE>
<CIRCULAR ARC CENTRE>
<CIRCULAR ARC CENTRE CLOSE>
<CIRCULAR ARC CENTRE BACKWARDS>
<ELLIPSE>
<ELLIPTICAL ARC>
<ELLIPTICAL ARC CLOSE>
<PIXEL ARRAY>

<LINE BUNDLE INDEX>
<LINE TYPE>
<LINE WIDTH>
<LINE COLOUR>
<MARKER BUNDLE INDEX>
<MARKER TYPE>
<MARKER SIZE>
<MARKER COLOUR>
<TEXT BUNDLE INDEX>
<TEXT FONT INDEX>
<TEXT PRECISION>
<TEXT FONT AND PRECISION>
<CHARACTER EXPANSION FACTOR>
<CHARACTER SPACING>
<TEXT COLOUR>
<CHARACTER HEIGHT>
<CHARACTER ORIENTATION>
<CHARACTER VECTORS>
<TEXT PATH>
<TEXT ALIGNMENT>
<CHARACTER SET INDEX>
<ALTERNATE CHARACTER SET INDEX>
<FILL BUNDLE INDEX>
<INTERIOR STYLE>
<FILL COLOUR>
<HATCH INDEX>
<PATTERN INDEX>
<EDGE BUNDLE INDEX>
<EDGE TYPE>
<EDGE WIDTH>
<EDGE COLOUR>
<EDGE VISIBILITY>
<FILL REFERENCE POINT>
<PATTERN TABLE>
<PATTERN SIZE>
<COLOUR TABLE>
<ASPECT SOURCE FLAGS>
<PICK IDENTIFIER>
<LINE REPRESENTATION>
<MARKER REPRESENTATION>
<TEXT REPRESENTATION>
<FILL REPRESENTATION>
<EDGE REPRESENTATION>
<PICK IDENTIFIER>
<DRAWING MODE>
<REOPEN SEGMENT>
<COPY SEGMENT>
<DELETE SEGMENT>
<DELETE ALL SEGMENT>
<RENAME SEGMENT>
<REDRAW ALL SEGMENTS>
<IMPLICIT SEGMENT REGENERATION MODE>
<INHERITANCE FILTER>
<SEGMENT TRANSFORM>
<SEGMENT VISIBILITY>
<SEGMENT HIGHLIGHTING>
<SEGMENT DISPLAY PRIORITY>
<SEGMENT DETECTABILITY>
<SEGMENT PICK PRIORITY>
<ESCAPE>
<MESSAGE>
<APPLICATION DATA>
<DRAWING SET>
<DRAWING PLUS CONTROL SET>

Annex G Formal Grammar of the GKSM Category

<to be added-this will be a subset of Annex F>

Annex H

H.1 The mapping of GKS functions to CGM elements

The table below shows a mapping between the GKS functions and CGM elements. The elements are those contained in the CGM as extended by this addendum.

Table XX The Mapping between GKS functions and CGM elements.

GKS Function	CGM Element
OPEN WORKSTATION	BEGIN METAFILE
	Metafile Descriptor
	{set metafile category}
	BEGIN PICTURE
	BEGIN PICTURE BODY
CLOSE WORKSTATION	END PICTURE
	END METAFILE
ACTIVATE WORKSTATION	Attribute Settings
	CLIP RECTANGLE
	Enable Output to Metafile
DEACTIVATE WORKSTATION	Disable Output to Metafile
CLEAR WORKSTATION	MAKE PICTURE CURRENT
	PREPARE VIEW SURFACE
	DELETE ALL SEGMENTS
REDRAW ALL SEGS ON WORKSTATION	MAKE PICTURE CURRENT
	PREPARE VIEW SURFACE
	REDRAW ALL SEGMENTS
UPDATE WORKSTATION	UPDATE
SET DEFERRAL STATE	DEFERRAL MODE
	IMPLICIT SEGMENT REGENERATION MODE
	{BNIG and BNIL map to BNI}
	{Maps to BNIG on interpretation}
MESSAGE	MESSAGE
ESCAPE	ESCAPE
POLYLINE	POLYLINE
POLYMARKER	POLYMARKER
TEXT	TEXT
FILL AREA	POLYGON
CELL ARRAY	CELL ARRAY
GDP	GDP
SET POLYLINE INDEX	LINE BUNDLE INDEX
SET LINETYPE	LINE TYPE
SET LINEWIDTH SCALE FACTOR	LINE WIDTH
SET POLYLINE COLOUR INDEX	LINE COLOUR
SET POLYMARKER INDEX	MARKER BUNDLE INDEX
SET MARKER TYPE	MARKER TYPE
SET MARKER SIZE SCALE FACTOR	MARKER SIZE
SET POLYMARKER COLOUR INDEX	MARKER COLOUR
SET TEXT INDEX	TEXT BUNDLE INDEX
SET TEXT FONT AND PRECISION	TEXT FONT INDEX
	TEXT PRECISION
SET CHARACTER EXPANSION FACTOR	CHARACTER EXPANSION FACTOR

SET CHARACTER SPACING	CHARACTER SPACING
SET TEXT COLOUR INDEX	TEXT COLOUR
SET CHARACTER HEIGHT	CHARACTER HEIGHT
SET CHARACTER UP VECTOR	CHARACTER ORIENTATION
SET TEXT PATHTEXT PATH	
SET TEXT ALIGNMENT	TEXT ALIGNMENT
SET FILL AREA INDEX	FILL BUNDLE INDEX
SET FILL AREA INTERIOR STYLE	INTERIOR STYLE
SET FILL AREA STYLE INDEX	HATCH INDEX
	PATTERN INDEX
SET FILL AREA COLOUR INDEX	FILL COLOUR
SET PATTERN SIZE	PATTERN SIZE
SET PATTERN REFERENCE POINT	FILL REFERENCE POINT
SET ASPECT SOURCE FLAGS	ASPECT SOURCE FLAGS
SET PICK IDENTIFIER	PICK IDENTIFIER
SET POLYLINE REPRESENTATION	LINE REPRESENTATION
SET TEXT REPRESENTATION	TEXT REPRESENTATION
SET FILL AREA REPRESENTATION	FILL REPRESENTATION
SET PATTERN REPRESENTATION	PATTERN TABLE
SET COLOUR REPRESENTATION	COLOUR TABLE
SET WINDOW	as current CGM Annex E
SET VIEWPORT	.
SET VIEWPORT INPUT PRIORITY	.
SELECT NORMALIZATION TRANSFORMATION	.
SET CLIPPING INDICATOR	CLIP RECTANGLE
SET WORKSTATION WINDOW	VDC EXTENT
SET WORKSTATION VIEWPORT	DEVICE VIEWPORT
CREATE SEGMENT	BEGIN SEGMENT
CLOSE SEGMENT	END SEGMENT
RENAME SEGMENT	RENAME SEGMENT
DELETE SEGMENT	DELETE SEGMENT
DELETE SEGMENT FROM WORKSTATION	DELETE SEGMENT
ASSOCIATE SEGMENT WITH WORKSTATION	BEGIN SEGMENT
	primitive and attributes
	END SEGMENT
COPY SEGMENT TO WORKSTATION	COPY SEGMENT
INSERT SEGMENT	COPY SEGMENT
SET SEGEMENT TRANSFORMATION	SEGMENT TRANSFORM
SET VISIBILITY	SEGMENT VISIBILITY
SET HIGHLIGHTING	SEGMENT HIGHLIGHTING
SET SEGEMENT PRIORITY	SEGMENT DISPLAY PRIORITY
	SEGMENT PICK PRIORITY
SET DETECTABILITY	SEGMENT DETECTABILITY
WRITE ITEM TO GKSM	APPLICATION DATA

On interpretation the reverse mapping will occur with item types being returned to the GKS application as indicated in Annex E of this addendum.

Information Processing Systems - Computer Graphics - Metafile for the Storage and Transfer of Picture Description Information

Part 2

Character Encoding

Addendum 1

November 1987

Page 10

Add the following to the end of sub-clause 5.3:

3/8 for Segment Control Elements and 3/9 for Segment Attribute Elements

Page 11

Add the following to table 1:

opcode	7bit coding	8 bit coding
BEGIN SEGMENT opcode	3/0	2/5
END SEGMENT opcode	3/0	2/6
METAFILE CATEGORY opcode	3/1	3/0
MAXIMUM VDC EXTENT opcode	3/1	3/1
SEGMENT PRIORITY EXTENT opcode	3/1	3/2
DEVICE VIEWPORT opcode	3/3	2/6
DEVICE V'PORT SPEC. MODE opcode	3/3	2/7
DEVICE VIEWPORT MAPPING opcode	3/3	2/8
DEFERRAL MODE opcode	3/3	2/9
MAKE PICTURE CURRENT opcode	3/3	2/10
PREPARE VIEW SURFACE opcode	3/3	2/11
UPDATE opcode	3/3	2/12
MODIFY FONT LIST opcode	3/3	2/13
MODIFY CHARACTER SET LIST opcode	3/3	2/14
BEGIN FIGURE opcode	3/3	2/15
END FIGURE opcode	3/3	3/0
NEW REGION opcode	3/3	3/1
IMPLICIT EDGE VISIBILITY opcode	3/3	3/2
SAVE PRIMITIVE ATTRIBUTES opcode	3/3	3/3
RESTORE PRIMITIVE ATTRIBUTES op.	3/3	3/4
DELETE ATTRIBUTE SET opcode	3/3	3/5
CIRCULAR ARC CENTRE BACKW op.	3/4	2/8
PIXEL ARRAY opcode	3/4	2/9
LINE REPRESENTATION opcode	3/5	2/8
MARKER REPRESENTATION opcode	3/5	2/9
TEXT REPRESENTATION opcode	3/5	3/12
FILL REPRESENTATION opcode	3/6	2/13
EDGE REPRESENTATION opcode	3/6	2/14
PICK IDENTIFIER opcode	3/6	3/2
DRAWING MODE opcode	3/6	3/3
REOPEN SEGMENT opcode	3/8	2/0
COPY SEGMENT opcode	3/8	2/1
DELETE SEGMENT opcode	3/8	2/2
DELETE ALL SEGMENTS opcode	3/8	2/3
RENAME SEGMENT opcode	3/8	2/4
REDRAW ALL SEGMENTS opcode	3/8	2/5
IMPLICIT SEGMENT REG. MODE	3/8	2/6
INHERITANCE FILTER opcode	3/8	2/7
SEGMENT TRANSFORM opcode	3/9	2/0
SEGMENT VISIBILITY opcode	3/9	2/1
SEGMENT HIGHLIGHTING opcode	3/9	2/2
SEGMENT DISPLAY PRIORITY opcode	3/9	2/3
SEGMENT DETECTABILITY opcode	3/9	2/4
SEGMENT PICK PRIORITY opcode	3/9	2/5

Page 27

Add the following sub-clause after sub-clause 6.12:

6.13 Attribute Set Name parameters

Attribute Set Name parameters are coded as integers (Basic format) at INTEGER PRECISION.

6.14 Device Point parameters

Device Point parameters are coded as integers (Basic format) at INTEGER PRECISION.

6.15 Segment Name parameters

Segment Name parameters are coded as integers (Basic format) at INTEGER PRECISION.

Page 31

The following form sub-clauses 8.1.6 and 8.1.7

8.1.6 BEGIN SEGMENT

<BEGIN-SEGMENT-opcode: 3/0 2/5>
<integer: segment-identifier>

8.1.7 END SEGMENT

<END-SEGMENT-opcode: 3/0 2/6>

Page 36

The following form sub-clauses 8.2.16 to 8.2.18

8.2.16 METAFILE CATEGORY

<METAFILE-CATEGORY-OPCODE: 3/1 3/0>
<enumerated: metafile category>

<enumerated: metafile category>	=	<integer: 0>	{cgm}
	|	<integer: 1>	{gksm}
	|	<integer: 2>	{cgmext1}

8.2.17 MAXIMUM VDC EXTENT

<MAXIUMUM VDC EXTENT-opcode: 3/1 3/1>
<point:first corner>
<point:second corner>

8.2.18 SEGMENT PRIORITY EXTENT

<SEGMENT PRIORITY EXTENT op-code: 3/1 3/2>
<integer: minimum-segment-priority value>
<integer:maximum-segment-priority-value>

Page 40

The following form sub-clauses 8.4.7 to 8.4.22

8.4.7 DEVICE VIEWPORT

<DEVICE-VIEWPORT-opcode: 3/3 2/6>

<device-point: first corner>)
<device-point : second corner>)

8.4.8 DEVICE VIEWPORT SPECIFICATION MODE

<DEVICE VIEWPORT SPECIFICATION MODE op-code: 3/3 2/7>
<enumerated:viewport-specification-unit>
<real: metric scale factor>

<enumerated: vieport-specification-unit>	=	<integer:0>	{fractn of def.dev vpt}
	|	<integer:1>	{mm with scale factor}
	|	<integer:2>	{physical dev units}

NOTE: Default not consistent with GKS

8.4.9 DEVICE VIEWPORT MAPPING

<DEVICE VIEWPORT MAPPING op-code: 3/3 2/8>
<enumerated: isotropy-flag>
<enumerated: horizontal-alignment-flag>
<enumerated: vertical alignment-flag>

<enumerated: isotropy-flag>	=	<integer:0>	{not forced}
	|	<integer:1>	{forced}
<enumerated: horizontal-alignment-flag>	=	<integer:0>	{left}
	|	<integer:1>	{centre}
	|	<integer:2>	{right}
<enumerated:vertical-alignment-flag>	=	<integer:0>	{bottom}
	|	<integer:1>	{centre}
	|	<integer:2>	{top}

8.4.10 SET DEFERRAL MODE

<SET-DEFERRAL-MODE-opcode: 3/3 2/9>
<enumerated: deferral mode>

<enumerated: deferral mode>	=	<integer: 0>	{asap}
	|	<integer: 1>	{bni}
	|	<integer: 2>	{asti}

8.4.11 MAKE PICTURE CURRENT

<MAKE PICTURE CURRENT op-code: 3/3 2/10>

8.4.12 PREPARE VIEW SURFACE>

<PREPARE VIEW SURFACE op-code: 3/3 2/11>
<enumerated: force-hard-copy-advance>

<enumerated:force-hard-copy-advance>	=	<integer:0>	{force hardcopy}
	|	<integer:1>	{conditional}

8.4.13 UPDATE

<UPDATE-opcode: 3/3 2/12>
<enumerated: update-regeneration-flag>

<enumerated: update-regeneration-flag>	=	<integer: 0>	{perform}
	|	<integer: 1>	{postpone}

8.4.14 MODIFY FONT LIST

<MODIFY FONT LIST-opcode: 3/3 2/13>
<integer: starting-index>
<font-declaration>+

<font-declaration>	=	<string: name of font>

8.4.15 MODIFY CHARACTER SET LIST

<MODIFY CHARACTER SET LIST-opcode: 3/3 2/14>
<integer: starting-index>
<character-set-declaration>+
<designation tail sequence>

<character-set-declaration>	= <integer:0>	{94-character G-set}
	| <integer:1>	{96-character G-set}
	| <integer:2>	{m'byte 94-charG-set}
	| <integer:3>	{m'byte 96-charG-set}
	| <integer:2>	{complete-code character set}}

8.4.16 BEGIN FIGURE

<BEGIN FIGURE-opcode 3/3 2/15>

8.4.17 END FIGURE

<END FIGURE-opcode: 3/3 3/0>

8.4.18 NEW REGION

<NEW REGION-opcode: 3/3 3/1>

8.4.19 IMPLICIT EDGE VISIBILITY

<IMPLICIT EDGE VISIBILITY-opcode: 3/3 3/2>
<enumerated: implicit-edge-visibility>

<enumerated: implicit-edge-visibility>	= <integer:0>	{off}
	| <integer:1>	{on}

8.4.20 SAVE PRIMITIVE ATTRIBUTES

<SAVE PRIMITIVE ATTRIBUTES-opcode: 3/3 3/3>
<integer: attribute-set-name>

8.4.21 RESTORE PRIMITIVE ATTRIBUTES

<RESTORE PRIMITIVE ATTRIBUTES-opcode: 3/3 3/4>
<integer: attribute-set-name>

8.4.22 DELETE ATTRIBUTE SET

<DELETE ATTRIBUTE SET-opcode: 3/3 3/5>
<integer: attribute-set-name>

Page 45

Add the following clauses after clause 8.5.19

8.5.20 CIRCULAR ARC CENTRE BACKWARDS

<CIRCULAR ARC CENTRE BACKWARDS-opcode: 3/4 2/8>
<point: centrepoint>
<VDC: DX_start>
<VDC: DY_start>
<VDC: DX_end>
<VDC: DY_end>
<VDC: radius>

8.5.21 PIXEL ARRAY

<PIXEL-ARRAY-opcode: 3/4 2/9>
<point: origin-point>
<integer: nx>
<integer: ny>
<integer: min-x-range>
<integer: max-x-range>
<integer: min-y-range>
<integer: max-y-range>
<integer: x-scale>
<integer: y-scale>

<local-colour-precision>	=	<default-colour-precision-indicator
	|	<number-of-bits>
<default-colour-precision-indicator>	=	<integer:0>
<number-of-bits>	=	<integer-positive>
<colour-list>	=	{as CGM Clause 8.5.9}

Page 54

The following form sub-clauses 8.6.36 to 8.6.42

8.6.36 LINE REPRESENTATION

<LINE REPRESENTATION-opcode: 3/5 2/8>
<integer: line-bundle-index>
<index: line-type>
<line-width-specifier>
<colour-specifier>

<integer: line-bundle-index>	=	<positive integer>	
<index: line-type>	=	<integer: 1>	{solid}
	|	<integer: 2>	{dash}
	|	<integer: 3>	{dot}
	|	<integer: 4>	{dash-dot}
	|	<integer: 5>	{dash-dot-dot}
	|	<integer: negative>	{private line type}
<line-width-specifier>	=	<real: line width-scale-factor> {if LINE WIDTH SPEC MODE is 'scaled'}	
	|	<VDC: line width> {if LINE WIDTH SPEC MODE IS 'absolute'}	
<colour-specifier>	=	<integer: colour index> {if COLOUR SELECTION MODE is 'indexed'}	
	|	<RGB> {if COL SELECTION MODE IS 'direct'}	
<integer: colour-index>	=	<non-negative integer>	

8.6.37 MARKER REPRESENTATION

<MARKER-REPRESENTATION-opcode: 3/5 2/9>
<integer: marker-bundle-index>
<index: marker-type>
<index: marker-type>
<marker-size-specifier>
<colour-specifier>

<integer: marker-bundle-index>	=	<positive integer>	
<index: marker-type>	=	<integer: 1>	{dot}
	|	<integer: 2>	{plus}
	|	<integer: 3>	{asterisk}
	|	<integer: 4>	{circle}
	|	<integer: 5>	{cross}
	|	<integer: negative>	{private marker type}

```
<marker-size-specifier>          =  <real: marker size-scale-factor>
                                    {if MARKER SIZE SPEC MODE is 'scaled'}
                                 |  <VDC: marker size>
                                    {if MARKER-SIZE SPEC MODE IS 'absolute'}
<colour-specifier>               =  <integer: colour index>
                                    {if COLOUR SELECTION MODE IS 'indexed'}
                                 |  <RGB>
                                    {if COLOUR SELEC MODE is 'direct'}
<integer: colour-index>          =  <non-negative integer>
```

8.6.38 TEXT REPRESENTATION

```
<TEXT-REPRESENTATION-opcode: 3/5 3/12>
<integer: text-bundle-index>
<integer: text-font-index>
<enumerated: text-precision>
<real: character-spacing>
<real: expansion-factor>
<colour-specifier>

<integer: text-bundle-index>     =  <positive integer>
<integer: text-font-index>       =  <positive integer>
<enumerated: text-precision>     =  <integer:0>        {string}
                                 |  <integer:1>        {character}
                                 |  <integer:2>        {stroke}
<real: expansion-factor>         =  <non-negative real>
```

8.6.39 FILL REPRESENTATION

```
<FILL-REPRESENTATION-opcode: 3/6 2/13>
<integer: fill-bundle-index>
<enumerated: interior-style>
<index: hatch-index>
<index: pattern-index>
<colour specifier>

<integer;fill-bundle-index>      =  <positive integer>
<enumerated: interior style>     =  <integer:0> {hollow
                                 |  <integer:1>        {solid
                                 |  <integer:2>        {pattern
                                 |  <integer:3>        {hatch
                                 |  <integer:4>        {empty
                                 |  <integer:negative>  {private style}

<index: hatch-index>             =  <integer:1>        {horizontal}
                                 |  <integer:2>        {vertical}
                                 |  <integer:3>        {positive slope}
                                 |  <integer:4>        {negative slope}
                                 |  <integer:5>        {horiz/vertical cross}
                                 |  <integer:6>        {positive/neg cross}
                                 |  <integer:negative>  {private styles}

<index; pattern-index>           =  <positive integer>
<colour specifier>               =  <integer:colour index>
                                    {if COLOUR SELECTION MODE is 'indexed'
                                 |  <RGB>
                                    {if COLOUR SELECTION MODE is 'direct'
```

8.6.40 EDGE REPRESENTATION

```
<EDGE-REPRESENTATION-opcode: 3/6 2/14>
<integer: edge-bundle-index>
<index: edge-type>
```

<edge-width-specifier>
<colour-specifier>

<integer: edge-bundle-index>	=	<positive integer>	
<index: edge-type>	=	<integer: 1>	{solid}
	|	<integer: 2>	{dash}
	|	<integer: 3>	{dot}
	|	<integer: 4>	{dash-dot}
	|	<integer: 5>	{dash-dot-dot}
	|	<integer: negative>	{private edge type}
<edge-width-specifier>	=	<real: edge-width-scale-factor>	
		{if EDGE WIDTH SPECI MODE is 'scaled'}	
	|	<VDC: edge width>	
		{if EDGE WIDTH SPEC MODE is 'absolute'}	
<colour-specifier>	=	<integer: colour-index>	
		{if COLOUR SELECTION MODE is 'indexed'}	
	|	<RGB>	
		{if COLOUR SELECTION MODE is 'direct'}	
<integer: colour-index>	=	<non-negative integer>	

8.6.41 PICK ID

<PICK-ID-opcode: 3/6 3/2>
<integer: pick-id>

8.6.42 DRAWING MODE

<DRAWING MODE-opcode: 3/6 3/3>
<integer:drawing-mode>

<integer:drawing-mode>	=	<integer:0>	{d' = 0}
	|	<integer:1>	{d' = s AND d}
	|	<integer:2>	{d' = s AND (NOT d)}
	|	<integer:3>	{d' = s}
	|	<integer:4>	{d' = (NOT s) AND d}
	|	<integer:5>	{d' = d}
	|	<integer:6>	{d' = s XOR d}
	|	<integer:7>	{d' = s OR d}
	|	<integer:8>	{d' = NOT (s OR d)}
	|	<integer:9>	{d' = NOT (s XOR d)}
	|	<integer:10>	{d' = NOT d}
	|	<integer:11>	{d' = s OR (NOT d)}
	|	<integer:12>	{d' = NOT s}
	|	<integer:13>	{d' = (NOT s) OR d}
	|	<integer:14>	{d' = NOT (s AND d)}
	|	<integer:15>	{d' = 1}

Page 56

The following forms sub-clauses 8.9 and 8.10

8.9 Segment Elements

8.9.1 REOPEN SEGMENT

<REPOPEN-SEGMENT-opcode: 3/8 2/0
<integer: segment-identifier>

8.9.2 COPY SEGMENT

<COPY-SEGMENT-opcode: 3/8 2/1>
<integer: segment-identifier>
<transformation matrix>

```
<transformation matrix>          =   <real: a11 >
                                 |   <real: a21 >
                                 |   <real: a12 >
                                 |   <real: a22 >
                                 |   <vdc : a13 >
                                 |   <vdc : a23 >
```

8.9.3 DELETE SEGMENT

```
<DELETE-SEGMENT-opcode: 3/8 2/2>
<integer: segment-identifier>
```

8.9.4 DELETE ALL SEGMENTS

```
<DELETE-ALL-SEGMENTS-opcode: 3/8 2/3>
```

8.9.5 RENAME SEGMENT

```
<RENAME-SEGMENT-opcode: 3/8 2/4>
<integer: old-segment-identifier>
<integer: new-segment-identifier>
```

8.9.6 REDRAW ALL SEGMENTS

```
<REDRAW-ALL-SEGMENTS-opcode: 3/8 2/5>
```

8.9.7 IMPLICIT SEGMENT REGENERATION MODE

```
<IMPLICIT-SEGMENT-REGENERATION-opcode: 3/8 2/6>
<enumerated: implicit-segment-regeneration-mode>
<enumerated: implicit-seg-regen-mode>   =   <integer:0>   {suppressed}
                                        |   <integer:1>   {allowed}
```

8.9.8 INHERITANCE FILTER

```
<INHERITANCE-FILTER-opcode: 3/8 2/7>
<enumerated: filter-selection-attribute-designator>
<enumerated: setting>
<enumerated: filter-selec-att-designator>   =   <integer:0>   {lines attributes}
                                            |   <integer:1>   {marker attrubute}
                                            |   <integer:2>   {text attributes}
                                            |   <integer:3>   {character attributes}
                                            |   <integer:4>   {fill attributes}
                                            |   <integer:5>   {edge attributes}
                                            |   <integer:6>   {pattern attributes}
                                            |   <integer:7>   {clip control}
                                            |   <integer:8>   {output control}
                                            |   <integer:9>   {all}
<enumerated: setting>   =   <state_list>
        |   <integer:1>   {segment}
```

8.10 Segment Attribute Elements

8.10.1 SEGMENT TRANSFORM

```
<SEGMENT-TRANSFORM-opcode: 3/9 2/0>
<integer: segment-identifier>
<transformation matrix>
<transformation matrix>          =   <real: a11 >
```

| <real: a21 >
| <real: a12 >
\ <real: a22 >
| <vdc : a13 >
| <vdc : a23 >

8.10.2 SEGMENT VISIBILITY

<SEGMENT-VISIBILITY-opcode: 3/9 2/1>
<integer: segment-identifier>
<enumerated: segment-visibility>

<enumerated: segment-visibility>	= <integer: 0>	{visible}
	| <integer: 1>	{invisibile}

8.10.3 SEGMENT HIGHLIGHTING

<SEGMENT-HIGHLIGHTING-opcode: 3/9 2/2>
<integer: segment-identifier>
<enumerated: segment-highlighting>

<enumerated: segment-highlighting>	= <integer: 0>	{normal}
	| <integer: 1>	{highlighted}

8.10.4 SEGMENT DISPLAY PRIORITY

<SEGMENT-PRIORITY-opcode: 3/9 2/3>
<integer: segment-identifier>
<integer: segment-display-priority>
<integer: segment-display-priority> = <positive integer>

8.10.5 SEGMENT DETECTABILITY

<SEGMENT-DETECTABILITY-opcode: 3/9 2/4>
<integer: segment-identifier>

<enumerated:segment-detectability>	= <integer: 0>	undetectable}
	| <integer: 1>	{detectable}

8.10.6 SEGMENT PICK PRIORITY

<SEGMENT-PICK-PRIORITY-opcode: 3/9 2/5>
<integer: segment-identifier>
<integer: pick-priority>
<integer: pick priority> = <positive integer>

ISO/TC97/SC24/N21

Information Processing Systems -

Computer Graphics -

Metafile for the Storage and Transfer of Picture Description Information

Part 3

Binary Encoding

Addendum 1

November 1987

Page 16

Add the following to table 1:

ASN	SI at integer precision (ip)	BASN {=ip/8}	ASNR {-2**(15) to 2**(15)-1}
DP	(I,I)	BDP {=2*BI}	DPR {-2**(15) to 2**(15)-1}
SN	SI at integer precision (ip)	BSN {=ip/8}	SNR {-2**(15) to 2**(15)-1}
VSP	(R,R)	BVSP {=2*BR}	VSPR = {RR}
	or		
	(I,I)	BVSP {=2*BI}	VSPR = {IR}
	or		
	DP	BVSP {=BDP}	VSPR = {DPR}

Page 17

Add the following to item 10):

h) metric scale factor

Page 19

Add the following to Table 2:

8 Segment Control elements
9 Segment Attribute elements

Page 20

Add the following to table 3:

BEGIN SEGMENT	6	SN	BSN	SNR	n/a
END SEGMENT	7	n/a	0	n/a	n/a

Code Description

6 BEGIN SEGMENT: has 1 parameter:
P1: (segment name) segment identifier

7 END SEGMENT: has no parameters

Page 21

Add the following to table 4:

METAFILE CATEGORY	16	E	BE	ER	0
MAXIMUM VDC EXTENT	17	2P	2BP	VDCR	VDC EXTENT
SEG PRIORITY EXTENT	18	2I	2BI	IR	0, 32767

(Note: Parameter range ER is not assigned to data type 'enumerated' in table 1, although used in table 8 for INTERIOR STYLE.)

Code Description

16 METAFILE CATEGORY: has 1 parameter:
P1: (enumerated) category: the following values are standardized:
0 cgm
1 gksm
2 cgmext1

17 MAXIMUM VDC EXTENT: has 2 parameters:
P1: (point) first point
P2: (point) second point

18 SEGMENT PRIORITY EXTENT: has 2 parameters:
P1: (integer) minimum segment priority value
P2: (integer) maximum segment priority value

Page 26

Add the following to table 6:

DEVICE VIEWPORT	7	2VSP	2BVSP	VSPR	see below
DEVICE VIEWPORT SPECIFICATION MODE	8	E,R	BE+BR	{0,1,2} RR	0,-
DEVICE VIEWPORT MAPPING	9	3E	3BE	{0,1} {0,1,2} {0,1,2}	1 0 0
DEFERRAL MODE	10	E	BE	{0,1,2}	0
MAKE PICTURE CURRENT	11	n/a	0	n/a	n/a
PREPARE VIEW SURFACE	12	E	BE	{0,1}	n/a
UPDATE	13	E	BE	{0,1}	n/a
MODIFY FONT LIST	14	IX,nS	BIX+nBS	+IXR,SR	n/a
MODIFY CHAR SET LIST	15	IX, n(E,S)	BIX+ n(BE+BS)	+IXR, {0..4},SR	n/a
BEGIN FIGURE	16	n/a	0	n/a	n/a
END FIGURE	17	n/a	0	n/a	n/a
NEW REGION	18	n/a	0	n/a	n/a
IMPLICIT EDGE VISIBILITY	19	E	BE	{0,1}	0
SAVE PRIMITIVE ATTS	20	ASN	BASN	ASNR	n/a
RESTORE PRIMITIVE ATTS	21	ASN	BASN	ASNR	n/a
DELETE ATTRIBUTE SET	22********				

Code Description

7 DEVICE VIEWPORT: has 2 parameters:
P1: (device point) first point
P2: (device point) second point

The default DEVICE VIEWPORT is the entire device view surface if the latter is rectangular, or the largest rectangular subset having the desired aspect ratio, if the view surface is not rectangular. The default is set so that the 'first point' is below and to the left of the 'second point' as seen by the viewer.

8 DEVICE VIEWPORT SPECIFICATION MODE: has 2 parameters:
P1: (enumerated) viewport specification units: valid values are:
0 fraction of default device viewport
1 millimetres with scale factor
2 physical device units
P2: (real) metric scale factor, ignored if P1=0 or P1=2

9 DEVICE VIEWPORT MAPPING: has 3 parameters:
P1: (enumerated) isotropy flag: valid values are:
0 not forced
1 forced
P2: (enumerated) horizontal alignment flag: valid values are:

0 left
1 centre
2 right
P3: (enumerated) vertical alignment flag: valid values are:
0 bottom
1 centre
2 top

10 DEFERRAL MODE: has 1 parameter:
P1: (enumerated) deferral mode: valid values are:
0 asti
1 bni
2 asap

11 MAKE PICTURE CURRENT: has no parameters

12 PREPARE VIEW SURFACE: has 1 parameter:
P1: (enumerated) force hardcopy advance: valid values are
0 force hardcopy
1 conditional

13 UPDATE: has 1 parameter:
P1: (enumerated) update regeneration flag: valid values are:
0 perform
1 postpone

14 MODIFY FONT LIST: has a variable parameter list:
P1: (index) starting font list index
P2-Pn: n font names (strings)

15 MODIFY CHARACTER SET LIST: has a variable parameter list:
P1: (index) starting character set list index
For each of the variable number of parameter pairs:
Pi: (enumerated) CHARACTER SET TYPE: valid values are:
0 94-character G-set
1 96-character G-set
2 94-character multibyte G-set
3 96-character multibyte G-set
4 complete code
negative for private use
P(i+1): (string) designation sequence tail; see part 1, 5.3.13.

16 BEGIN FIGURE: has no parameters

17 END FIGURE: has no parameters

18 NEW REGION: has no parameters

19 IMPLICIT EDGE VISIBILITY: has 1 parameter:
P1: (enumerated) implicit edge visibility: valid values are:
0 off
1 on

20 SAVE PRIMITIVE ATTRIBUTES: has 1 parameter:
P1: (attribute set name) attribute set name

21 RESTORE PRIMITIVE ATTRIBUTES: has 1 parameter:
P1: (attribute set name) attribute set name

22 DELETE ATTRIBUTE SET: has 1 parameter:
P1: (attribute set name) attribute set name

Page 28

Add the following to table 7:

CIRCULAR ARC CENTRE BACKWARDS	20	P,4VDC, VDC	BP+4BVDC+ BVDC	VDCR,VDCR, ++VDCR	n/a
PIXEL ARRAY	21	P,6I, 2I, CLIST	BP+6BI+ 2BI+ nBCO	VDCR,IR, +IR, COR	n/a

Code Description

20 CIRCULAR ARC CENTRE BACKWARDS: has 6 parameters:
P1: (point) centre of circle
P2: (vdc) delta X for start vector
P3: (vdc) delta Y for start vector
P4: (vdc) delta X for end vector
P5: (vdc) delta Y for end vector
P6: (vdc) radius of circle

21 PIXEL ARRAY: has 10 parameters:
P1 : (point) origin point
P2 : (integer) nx
P3 : (integer) ny
P4 : (integer) minimum of valid x-range
P5 : (integer) maximum of valid x-range
P6 : (integer) minimum of valid y-range
P7 : (integer) maximum of valid y-range
P8 : (integer) x scale
P9 : (integer) y scale
P10: (colour list) array of colour specifiers

Page 33

Add the following to table 8:

LINE REPRESENTATION	36	2IX, (VDC or R),CO	2BIX+ (BVDC or BR)+BCO	+IXR,IXR, ++VDCR or ++RR,COR	n/a
MARKER REPRESENTATION	37	2IX, (VDC or R),CO	2BIX+ (BVDC or BR)+BCO	+IXR,IXR, ++VDCR or ++RR,COR	n/a
TEXT REPRESENTATION	38	2IX, E, 2R,CO	2BIX+ BE+ 2BR+BCO	+IXR, {0,1,2}, RR,COR	n/a
FILL REPRESENTATION	39	IX, E,CO, 2IX	BIX+ BE+BCO+ 2BIX	+IXR, {0..4},COR, IXR,+IXR	n/a
EDGE REPRESENTATION	40	2IX, (VDC or R),CO	2BIX+ (BVDC or BR)+BCO	+IXR,IXR, ++VDCR or ++RR,COR	n/a
PICK IDENTIFIER	41	I	BI	IR	n/a
DRAWING MODE	42	I	BI	++IR	3

Code Description

36 LINE REPRESENTATION: has 4 parameters:
P1: (index) line bundle index
P2: (index) line type: the following values are standardized:
1 solid
2 dash
3 dot
4 dash-dot

5 dash-dot-dot
negative for private use
P3: (vdc or real) absolute line width or line width scale factor
P4: (colour) line colour: its form depends on COLOUR SELECTION MODE.

37 MARKER REPRESENTATION: has 4 parameters:
P1: (index) marker bundle index
P2: (index) marker type: the following values are standardized:
1 dot
2 plus
3 asterisk
4 circle
5 cross
negative for private use
P3: (vdc or real) absolute marker width or marker size scale factor
P4: (colour) marker colour: its form depends on COLOUR SELECTION MODE.

38 TEXT REPRESENTATION: has 6 parameters:
P1: (index) text bundle index
P2: (index) text font index
P3: (enumerated) text precision: valid values are:
0 string
1 character
2 stroke
P4: (real) character spacing
P5: (real) character expansion factor
P6: (colour) text colour; its form depends on COLOUR SELECTION MODE

39 FILL REPRESENTATION: has 5 parameters:
P1: (index) fill area bundle
P2: (index) interior style: valid values are:
P3: (index) hatch index: the following values are standardized:
1 horizontal
2 vertical
3 positive slope
4 negative slope
5 combined vertical and horizontal slant
6 combined left and right slant
negative for private use
P4: (index) pattern index
P5: (colour) fill colour: its form depends on COLOUR SELECTION MODE

40 EDGE REPRESENTATION: has 4 parameters:
P1: (index) edge bundle index
P2: (index) edge type: the following values are standardized:
1 solid
2 dash
3 dot
4 dash-dot
5 dash-dot-dot
negative for private use
P3: (vdc or real) absolute edge width or line width scale factor
P4: (colour) edge colour: its form depends on COLOUR SELECTION MODE.

41 PICK IDENTIFIER: has 1 parameter:
P1: (integer) pick identifier

42 DRAWING MODE: has 1 parameter:
P1: (integer) drawing mode: valid values are:
0 d'=0
1 d'=s AND d
2 d'=s AND (NOT d)
3 d'=s
4 d'=(NOT s) AND d

5 d'=d
6 d'=s XOR d
7 d'=s OR d
8 d'=NOT (s OR d)
9 d'=NOT (s XOR d)
10 d'=NOT d
11 d'=s OR (NOT d)
12 d'=NOT s
13 d'=(NOT s) OR d
14 d'=NOT (s AND d)
15 d'=1
(d' = resulting destination bit value,
d = original destination bit value,
s = original source bit value)

Page 39

The following forms sub-clause 7.10

7.10 Segment Control Elements

Table 11 - Encoding of segment control elements

Element Class 8	El Id	Param Type	Parameter List Length	Parameter Range	Default
REOPEN SEGMENT	1	SN	BSN	SNR	n/a
COPY SEGMENT	2	SN,4R, 2VDC	BSN+4BR+ 2BVDC	SNR,RR, VDCR	n/a
DELETE SEGMENT	3	SN	BSN	SNR	n/a
DELETE ALL SEGMENTS	4	n/a	0	n/a	n/a
RENAME SEGMENT	5	2SN	2BSN	SNR	n/a
REDRAW ALL SEGMENTS	6	n/a	0	n/a	n/a
IMPLICIT SEGMENT REGENERATION MODE	7	E	BE	{0,1}	0
INHERITANCE FILTER	8	nE,E	(n+1)BE	{0..9},{0,1}	-,1

Code Description

1 REOPEN SEGMENT: has 1 parameter:
P1: (segment name) segment identifier

2 COPY SEGMENT: has 7 parameters:
P1: (segment name) segment identifier
The remaining 6 parameters are components of a 3x2 matrix of the form:

|P2 P3 P6|
|P4 P5 P7|

where
P2: (real) x scale component
P3: (real) x rotation component
P4: (real) y rotation component
P5: (real) y scale component
P6: (vdc) x translation component
P7: (real) y translation component

3 DELETE SEGMENT: has 1 parameter:
P1: (segment name) segment identifier

4 DELETE ALL SEGMENTS: has no parameters

5 RENAME SEGMENT: has 2 parameters:

P1: (segment name) old segment identifier
P2: (segment name) new segment identifier

6 REDRAW ALL SEGMENTS: has no parameters

7 IMPLICIT SEGMENT REGENERATION MODE: has 1 parameter:
P1: (enumerated) implicit segment regeneration mode: valid values are:
0 suppressed
1 allowed

8 INHERITANCE FILTER: has up to 11 parameters, corresponding to each of the 10 inheritance filter function groups plus the setting:
(enumerated) filter selection attribute designator: valid values are:
0 line attributes
1 marker attributes
2 text attributes
3 character attributes
4 fill attributes
5 edge attributes
6 pattern attributes
7 clip control
8 output control
9 all

(enumerated) setting: valid values are:
0 state_list
1 segment

Page 39

The following forms sub-clause 7.11

7.11 Segment Attribute Elements

Table 12 - Encoding of segment attribute elements

Element Class 9	El Id	Param Type	Parameter List Length	Parameter Range	Default
SEGMENT TRANSFORM	1	SN,4R, 2VDC	BSN+4BR+ 2BVDC	SNR,RR, VDCR	n/a,1,0,0,1, 0,0
SEGMENT VISIBILITY	2	SN,E	BSN+BE	SNR,{0,1}	n/a,0
SEGMENT HIGHLIGHTING	3	SN,E	BSN+BE	SNR,{0,1}	n/a,0
SEG DISPLAY PRIORITY	4	SN,I	BSN+BI	SNR,IR	n/a,see below
SEGMENT DETECTABILITY	5	SN,E	BSN+BE	SNR,{0,1}	n/a,0
SEGMENT PICK PRIORITY	6	SN,I	BSN+BI	SNR,IR	n/a,see below

Code Desription

1 SEGMENT TRANSFORM: has 7 parameters:
P1: (segment name) segment identifier
The remaining 6 parameters are components of a 3x2 matrix of the form:

|P2 P3 P6|
|P4 P5 P7|
where
P2: (real) x scale component
P3: (real) x rotation component
P4: (real) y rotation component
P5: (real) y scale component
P6: (vdc) x translation component
P7: (vdc) y translation component

2 SEGMENT VISIBILITY: has 2 parameters:
P1: (segment name) segment identifier
P2: (enumerated) visibility: valid values are:
0 visible
1 invisible

3 SEGMENT HIGHLIGHTING: has 2 parameters:
P1: (segment name) segment identifier
P2: (enumerated) highlighting: valid values are:
0 normal
1 highlighted

4 SEGMENT DISPLAY PRIORITY: has 2 parameters:
P1: (segment name) segment identifier
P2: (integer) segment display priority
The default of the segment display priority is equal to the minimum segment priority value (see sub-clause 7.3)

5 SEGMENT DETECTABILITY: has 2 parameters:
P1: (segment name) segment identifier
P2: (enumerated) detectability: valid values are:
0 undetectable
1 detectable

6 SEGMENT PICK PRIORITY: has 2 parameters:
P1: (segment name) segment identifier
P2: (integer) segment pick priority
The default of the segment display priority is equal to the minimum segment priority value (see sub-clause 7.3)

ISO/TC97/SC24/N22

Information Processing Systems -

Computer Graphics -

Metafile for the Storage and Transfer of Picture Description Information

Part 4

Clear Text Encoding

Addendum 1

November 1987

Page 11

Add the following to the end of sub-clause 5.3.5

ASN ::= <I> {attribute set name}

SN ::= <I> {segment name}

VC ::= <R>|<I> {depending on DEV VIEWPORT SPEC MODE}

VSPOINTREC ::= <VC><SEP><VC>

VSP ::= <VSPOINTREC>|< <LEFT PAREN><OPTSEP><VSPOINTREC><OPTSEP> <RIGHT PAREN> >

{COORDINATE in viewport specification space. Parentheses are optional. If they are used, they shall group exactly two real or integer numbers, depending on DEVICE VIEWPORT SPECIFICATION MODE. The parenthesized form is intended to aid readability of the metafile}

TM ::= <R><SEP><R><SEP><VDC><SEP><R><SEP><R><SEP><VDC>

{2*3 real transformation matrix in row-major order}

Page 12

Add the following to the end of sub-clause 5.4.3

ALL
CATEGORY
CONDITIONAL
COPY
DELETE
DEVICE
DISPLAY
DRAWING
FIGURE
FILTER
FORCE
FRACTION
HARDCOPY
MAKE
NEW
PICK
PIXEL
PREPARE
REDRAW
REGION
REOPEN
SAVE
SURFACE
UNITS
UPDATE
VIEW

Page 12

Add the following to the end of sub-clause 5.4.4

ATTRIBUTE(S)	ATTR
BACKWARDS	BACK
CURRENT	CUR
DEFERRAL	DEFER
DETECTABLE	DET
DETECTABILITY	DET
HIGHLIGHTING	HIGHLIGHT
IDENTIFIER	ID
IMPLICIT	IMPL
INHERITANCE	INH
MAPPING	MAP
MILLIMETER	MM
MODIFY	MOD
PHYSICAL	PHY
PRIMITIVE	PRIM
PRIORITY	PRI
REGENERATION	REGEN
REPRESENTATION	REP

RESTORE	RES
SEGMENT	SEG
TRANSFORM	TRANS
UNDETECTABLE	UNDET

Page 14

Add the following to the end of sub-clause 5.4.5

BEGIN SEGMENT	BEGSEG
END SEGMENT	ENDSEG
METAFILE CATEGORY	MFCATEGORY
MAXIMUM VDC EXTENT	MAXVDCEXT
SEGMENT PRIORITY EXTENT	SEGPRIEXT
DEVICE VIEWPORT	DEVICEVIEWPORT
DEVICE VIEWPORT SPECIFICATION MODE	DEVICEVIEWPORTMODE
DEVICE VIEWPORT MAPPING	DEVICEVIEWPORTMAP
DEFERRAL MODE	DEFERMODE
MAKE PICTURE CURRENT	MAKEPICCUR
PREPARE VIEW SURFACE	PREPAREVIEWSURFACE
UPDATE	UPDATE
MODIFY FONT LIST	MODFONTLIST
MODIFY CHARACTER SET LIST	MODCHARSETLIST
BEGIN FIGURE	BEGFIGURE
END FIGURE	ENDFIGURE
NEW REGION	NEWREGION
IMPLICIT EDGE VISIBILITY	IMPLEDGEVIS
SAVE PRIMITIVE ATTRIBUTES	SAVEPRIMATTR
RESTORE PRIMITIVE ATTRIBUTES	RESPRIMATTR
DELETE ATTRIBUTE SET	DELETEATTRSET
CIRCULAR ARC CENTRE BACKWARDS	ARCCTRBACK
PIXEL ARRAY	PIXELARRAY
LINE REPRESENTATION	LINEREP
MARKER REPRESENTATION	MARKERREP
TEXT REPRESENTATION	TEXTREP
FILL REPRESENTATION	FILLREP
EDGE REPRESENTATION	EDGEREP
PICK IDENTIFIER	PICKID
DRAWING MODE	DRAWINGMODE
REOPEN SEGMENT	REOPENSEG
COPY SEGMENT	COPYSEG
DELETE SEGMENT	DELETESEG
DELETE ALL SEGMENTS	DELETEALLSEG
RENAME SEGMENT	RENAMESEG
REDRAW ALL SEGMENTS	REDRAWALLSEG
MPLICIT SEGMENT REGENERATION MODE	IMPLSEGREGENMODE
INHERITANCE FILTER	INHFILTER
SEGMENT TRANSFORM	SEGTRANS
SEGMENT VISIBILITY	SEGVIS
SEGMENT HIGHLIGHTING	SEGHIGHLIGHT
SEGMENT DISPLAY PRIORITY	SEGDISPLAYPRI
SEGMENT DETECTABILITY	SEGDET
SEGMENT PICK PRIORITY	SEGPICKPRI

```
                                <TERM>

DEFERRAL MODE          ::=  DEFERMODE
                                <SOFTSEP>
                            <TERM>
```

Note: GKS has BNIG and BNIL instead of BNI. In addition the chosen CGI default is not consistent with GKS!

```
MAKE PICTURE CURRENT   ::=  MAKEPICCUR
                            <TERM>

PREPARE VIEW SURFACE   ::=  PREPAREVIEWSURFACE
                                <SOFTSEP>
                                <FORCEHARDCOPY|CONDITIONAL>
                            <TERM>

UPDATE                 ::=  UPDATE
                                <SOFTSEP>
                                <PERFORM|POSTPONE>
                            <TERM>

MODIFY FONT LIST       ::=  MODFONTLIST
                                <SOFTSEP>
                                <I:STARTINDEX>          {positive}
                                <SEP>
                                <S:FONTNAME>
                                <SEP> <S:FONTNAME>*
                            <TERM>

MODIFY CHARACTER SET LIST  ::=  MODCHARSETLIST
                                <SOFTSEP>
                                <I:STARTINDEX> {positive}
                                <CHARSETDESIGNATOR>+
                            <TERM>

CHARSETDESIGNATOR      ::=  <SEP>
                                <STD94|
                                STD96|
                                STD94MULTIBYTE|
                                STD96MULTIBYTE|
                                COMPLETECODE>
                                <<OPTSEP>|<HARDSEP>>
                                <S:TAIL>

BEGIN FIGURE           ::=  BEGFIGURE <TERM>

END FIGURE             ::=  ENDFIGURE <TERM>

NEW REGION             ::=  NEWREGION <TERM>

IMPLICIT EDGE VISIBILITY   ::=  IMPLEDGEVIS
                                <SOFTSEP>
                                <OFF|ON>
                            <TERM>

SAVE PRIMITIVE ATTRIBUTES  ::=  SAVEPRIMATTR
                                <SOFTSEP>
                                <ASN:ATTRIBUTESETNAME>
                            <TERM>

RESTORE PRIMITIVE ATTS     ::=  RESPRIMATTR
                                <SOFTSEP>
                                <ASN:ATTRIBUTESETNAME>
```

Page 15

Add the following to the end of sub-clause 6.2

```
BEGIN SEGMENT          ::= BEGSEG
                               <SOFTSEP>
                               <I:SEGID>
                           <TERM>

END SEGMENT            ::= ENDSEG
                           <TERM>
```

Page 17

Add the following to the end of sub-clause 6.3

```
METAFILE CATEGORY      ::= MFCATEGORY
                               <SOFTSEP>
                               <CGM|GKSM|CGMEXT1>
                           <TERM>

MAXIMUM VDC EXTENT     ::= MAXVDCEXT
                               <SOFTSEP>
                               <P:FIRSTCORNER>
                               <SEP>
                               <P:SECONDCORNER>
                           <TERM>

SEGMENT PRIORITY EXTENT ::= SEGPRIEXT
                               <SOFTSEP>
                               <I:MINSEGPRI>
                               <SEP>
                               <I:MAXSEGPRI>
                           <TERM>
```

Page 19

Add the following to the end of sub-clause 6.5

```
DEVICE VIEWPORT        ::= DEVICEVIEWPORT
                               <SOFTSEP>
                               <VSP:FIRSTCORNER>
                               <SEP>
                               <VSP:SECONDCORNER>
                           <TERM>

DEVICE VIEWPORT SPEC MODE ::= DEVICEVIEWPORTMODE
                               <SOFTSEP>
                               <FRACTION|MM|PHYDEVICEUNITS>
                               <SEP>
                               <R:SCALEFACTOR>
                           <TERM>
```

Note: The chosen CGI default is not consistent with GKS

```
DEVICE VIEWPORT MAPPING ::= DEVICEVIEWPORTMAP
                               <SOFTSEP>
                               <NOTFORCED|FORCED>
                               <SEP>
                               <LEFT|CTR|RIGHT>
                               <SEP>
                               <BOTTOM|CTR|TOP>
```

```
                                <TERM>

DELETE ATTRIBUTE SET        ::= DELETEATTRSET
                                    <SOFTSEP>
                                    <ASN:ATTRIBUTESETNAME>
                                <TERM>
```

Page 24

Add the following to the end of sub-clause 6.6

```
CIRCULAR ARC CENTRE BACKW   ::= ARCCTRBACK
                                    <CTRARCSPEC>
                                    <TERM>

PIXEL ARRAY                 ::= PIXELARRAY
                                    <SOFTSEP>
                                    <P:ORIGIN_POINT>
                                    <SEP>
                                    <I:NX>
                                    <SEP>
                                    <I:NY>
                                    <SEP>
                                    <I:MIN_X_RANGE>
                                    <SEP>
                                    <I:MAX_X_RANGE>
                                    <SEP>
                                    <I:MIN_Y_RANGE>
                                    <SEP>
                                    <I:MAX_Y_RANGE>
                                    <SEP>
                                    <I:X_SCALE>
                                    <SEP>
                                    <I:Y_SCALE>
                                    <SEP>
                                    <LOCLCOLRSPEC>
                                    <CELLROW>*
                                <TERM>
```

Page 28

Add the following to the end of sub-clause 6.7

```
LINE REPRESENTATION         ::= LINEREP
                                    <SOFTSEP>
                                    <I:BUNDLEINDEX> {positive}
                                    <SEP>
                                    <I:LINETYPE>
                                            {1=solid, 2=dash
                                            3=dot, 4=dash-dot
                                            5=dash-dot-dot
                                            <0 implement'n dependent}
                                    <SEP>
                                    <V:LINEWIDTH> {non-negative}
                                    <SEP>
                                    <K:LINECOLR>
                                <TERM>

MARKER REPRESENTATION       ::= MARKERREP
                                    <SOFTSEP>
                                    <I:BUNDLEINDEX> {positive}
                                    <SEP>
                                    <I:MARKERTYPE>
```

```
                    {1=dot, 2=plus
                    3=asterisk, 4=circle
                    5=cross (x)
                    <0 implement'n dependent}
            <SEP>
            <V:MARKERSIZE> {non-negative}
            <SEP>
            <K:MARKERCOLR>
        <TERM>

TEXT REPRESENTATION     ::= TEXTREP
            <SOFTSEP>
            <I:BUNDLEINDEX> {positive}
            <SEP>
            <I:FONTINDEX> {positive}
            <SEP>
            <STRING|CHAR|STROKE>
            <SEP>
            <R:SPACING>
            <SEP>
            <R:FACTOR>
            <SEP>
            <K:TEXTCOLR>
        <TERM>

FILL REPRESENTATION     ::= FILLREP
            <SOFTSEP>
            <I:BUNDLEINDEX> {positive}
            <SEP>
            <HOLLOW|SOLID|PAT|HATCH|EMPTY>
            <SEP>
            <I:HATCHINDEX>
                    {1=horizontal,2=vertical
                    3=positive slope
                    4=negative slope
                    5=horiz/vert cross
                    6=+/- slope cross
                    <0 implement. dependent
            <SEP>
            <I:PATINDEX> {positive}
            <SEP>
            <K:FILLCOLR>
        <TERM>

EDGE REPRESENTATION     ::= EDGEREP
            <SOFTSEP>
            <I:BUNDLEINDEX> {positive}
            <SEP>
            <I:EDGETYPE>
                    {1=solid, 2=dash
                    3=dot, 4=dash-dot
                    5=dash-dot-dot
                    <0 implement'n dependent}
            <SEP>
            <V:EDGEWIDTH>
            <SEP>
            <K:EDGECOLR>
        <TERM>

PICK IDENTIFIER         ::= PICKID
            <SOFTSEP>
            <I:SEGID>
        <TERM>
```

```
DRAWING MODE          ::= DRAWINGMODE
                              <SOFTSEP>
                              <I:DRAWINGMODE>
                                      {0  d'=0
                                      1  d'=s AND d
                                      2  d'=s AND (NOT d)
                                      3  d'=s
                                      4  d'=(NOT s) AND d
                                      5  d'=d
                                      6  d'=s XOR d
                                      7  d'=s OR d
                                      8  d'=NOT (s OR d)
                                      9  d'=NOT (s XOR d)
                                      10  d'=NOT d
                                      11  d'=s OR (NOT d)
                                      12  d'=NOT s
                                      13  d'=(NOT s) OR d
                                      14  d'=NOT (s AND d)
                                      15  d'=1}
                        <TERM>
```

Page 29

The following forms sub-clause 6.10

6.10 Encoding Segment Control Elements

```
REOPEN SEGMENT        ::= REOPENSEG
                              <SOFTSEP>
                              <SN:SEGID>
                        <TERM>

COPY SEGMENT          ::= COPYSEG
                              <SOFTSEP>
                              <SN:SEGID>
                              <SEP>
                              <TM:TRANSMATRIX>
                        <TERM>

DELETE SEGMENT        ::= DELETESEG
                              <SOFTSEP>
                              <SN:SEGID>
                        <TERM>

DELETE ALL SEGMENTS   ::= DELETEALLSEG <TERM>

RENAME SEGMENT        ::= RENAMESEG
                              <SOFTSEP>
                              <SN:OLDSEGID>
                              <SEP>
                              <SN:NEWSEGID>
                        <TERM>

REDRAW ALL SEGMENTS   ::= REDRAWALLSEG <TERM>

IMPLICIT SEG REGEN. MODE ::= IMPLSEGREGENMODE
                              <SOFTSEP>
                              <SUPPRESSED|ALLOWED>
                        <TERM>

INHERITANCE FILTER    ::= INHFILTER
                              <SOFTSEP>
```

```
                    <LINEATTR|
                    MARKERATTR|
                    TEXTATTR|
                    CHARATTR|
                    FILLATTR|
                    EDGEATTR|
                    PATATTR|
                    CLIPCONTROL|
                    OUTPUTCONTROL|
                    ALL>
                    <SEP>
                    <STATELIST|SEGMENT>
                <TERM>
```

Page 29

The following forms sub-clause 6.11

6.11 Encoding Segment Attribute Elements

```
SEGMENT TRANSFORM        ::= SEGTRAN
                                 <SOFTSEP>
                                 <SN:SEGID>
                                 <SEP>
                                 <TM:TRANSMATRIX>
                             <TERM>

SEGMENT VISIBILITY       ::= SEGVIS
                                 <SOFTSEP>
                                 <SN:SEGID>
                                 <SEP>
                                 <VIS|INVIS>
                             <TERM>

SEGMENT HIGHLIGHTING     ::= SEGHIGHLIGHT
                                 <SOFTSEP>
                                 <SN:SEGID>
                                 <SEP>
                                 <NORMAL|HIGHLIGHTED>
                             <TERM>

SEGMENT DISPLAY PRIORITY ::= SEGDISPLAYPRI
                                 <SOFTSEP>
                                 <SN:SEGID>
                                 <SEP>
                                 <I:DISPLAYPRIORITY>
                             <TERM>

SEGMENT DETECTABILITY    ::= SEGDET>
                                 <SOFTSEP>
                                 <SN:SEGID>
                                 <SEP>
                                 <UNDET|DET>
                             <TERM>

SEGMENT PICK PRIORITY    ::= SEGPICKPRI
                                 <SOFTSEP>
                                 <SN:SEGID>
                                 <SEP>
                                 <I:PICKPRIORITY>
                             <TERM>
```

APPENDIX 3

U.S. PDAD BALLOT RESPONSES

This page left intentionally blank.

"Computer Graphics"

N727

Date: April 1988

Title: US Responses to ISO 8632-1 to 4/PDAD 1

Source: US

Project: JTC1.24.5.5
JTC1.24.5.6
JTC1.24.5.7
JTC1.24.5.8

Distribution: P, O, L Members
Secretariat JTC 1, CS, SC 24 Chairman and WG-Convenors

Date of Submission: 1988-04-12

RECEIVED
APR 18 1988
INFORMATION SYSTEMS

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION · МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ · ORGANISATION INTERNATIONALE DE NORMAL

Secretariat address
DIN · Burggrafenstrasse 4-10 · D-1000 Berlin 30

Telephone
(030) 2601-1

Telex
184 273 din d

Telegrams
DEUTSCHNORMEN
BERLIN

U.S. Response to CGM DPAD 1 Part 1

The U. S. disapproves of Part 1 of CGM PDAD 1 (SC24/N19), with the following comments. The part 1 comments are organized in 3 sections: The first section contains all comments except segmentation and formal grammar; The second section contains segmentations comments; And the third section contains formal grammar comments.

Technical Comments

1/T1) Global comment: How much discussion of errors is appropriate for the metafile? Most errors are a function of the semantics of the interpreter. Since GKS thoroughly specifies its errors and associated reactions to them, it is not necessary to repeat this in this document.

1/T2) Page 2: Is there supposed to be a 'GKSM' shorthand for the Metafile Elements List?

1/T3) DEVICE VIEWPORT MAPPING is not needed in the GKSM0 set, since GKS only requires the default of (isotropic, left, bottom).

1/T4) 4.12.4.7 and page 31, SEGMENT TRANSFORM: Needs to be rewritten to reflect output of Valbonne CGI that the locus is transformed, not just the reference points.

1/T5) A complete description of the two mutually exclusive ways of mapping VDC to the device (using SCALING MODE 'metric' or using DEVICE VIEWPORT), along with the statement about using the most recently specified if both are permitted in the metafile category, should be in clause 4, not buried in Annex D. Pictures are critical here!

1/T6) A section is needed describing how transformation affects primitives and attributes. Page 10, section 4.12.4.7, paragraphs 3 and 4: These paragraphs need to be reworded to reflect the change to CGI regarding segment transformations applying to the locus of graphical objects, rather than reference points. State the appearance of CIRCLE, RECTANGLE, PIXEL ARRAY, and RESTRICTED TEXT under locus transformation.

1/T7) RENAME SEGMENT and DELETE SEGMENT should not be permitted in global segments. These do not appear to be necessary or useful.

1/T8) COLOUR SELECTION MODE, LINE WIDTH SPECIFICATION MODE, EDGE WIDTH SPECIFICATION MODE, and MARKER SIZE SPECIFICATION MODE should be permitted in global and local segments as well as picture open. Even though GKS does not make use of these functions, a non-GKS client could assemble a "library" of global segments from different metafiles, to be referenced in the pictures of the metafile. CGI has concluded that there is no difficulty in mixing these modes within a single frame or session.

1/T9) The CGM currently does not specify the order of the metafile descriptor elements, although the formal grammar seems to have more restrictions in this regard than the body of the standard would indicate. We believe it would be a good idea if the first four MD elements were the METAFILE VERSION, METAFILE ELEMENT LIST, METAFILE CATEGORY, and METAFILE DESCRIPTION. If the Addendum does make this

a requirement, it cannot be extended to basic category CGM for reasons of backward compatibility (although it can be noted as "advice to generators").

Note: This does not match the formal grammar of the CGM-IS text.

1/T10) In Table 3(a), there are a number of elements permitted in states FO and FC that should not be. These are: CLIP RECTANGLE and CLIP INDICATOR, DEVICE VIEWPORT SPECIFICATION MODE, PREPARE VIEW SURFACE, and UPDATE. DEFERRAL MODE and MAKE PICTURE CURRENT should probably also be deleted from those states, as they make no sense - there is no rendering or updatable action until the completion of the figure definition.

1/T11) In Table 3(a), END FIGURE and NEW REGION should only be permitted in FO and FC.

1/T12) In Table 3(a), APPEND TEXT should be permitted only in state TO.

1/T13) BEGIN METAFILE should appear in the state table (Table 3(a)) or in an associated note.

1/T14) Annex D needs to be completed for the new elements. In particular, what is a vector device recommended to do with a PIXEL ARRAY?

1/T15) page 33: The DEVICE VIEWPORT should default to (0,0)(1,1) rather than implementation dependent (since the default VIEWPORT SPECIFICATION UNITS is "fraction of display surface). Taken along with the default DEVICE VIEWPORT MAPPING of 'isotropic', this default viewport produces the largest inscribed square which fits on the workstation display surface.

1/T16) page 33: PICK IDENTIFIER's default should be zero.

1/T17) 7.5: It is not appropriate to specify conformance in terms of the formal grammar, which is not part of the standard. "Grammar" would suffice here.

1/T18) E.8.2 needs a corresponding paragraph for generating the correct CGM font information from GKS.

1/T19) 5.10.10: The order of the enumerated parameters should be (invisible, visible). This conforms to the "null value rule". Also, the CGM already has these enumerated types bound in this order for the edge control of POLYGON SET.

1/T20) 5.10.13: The order of the enumerated parameters should be (undetectable, detectable), by the "null value rule".

1/T21) COPY SEGMENT should be permitted in state PO, to accomplish what GKS does with INSERT SEGMENT (copying a segment from WISS to nonretained data). Table 3(a) would need to be updated as well.

Note: This is a resolved CGI issue out of the San Diego meeting.

1/T22) PIXEL ARRAY should have a local colour precision parameter (like CELL ARRAY and PATTERN TABLE), to permit compaction of the colour list. (Liaison with CGI is required before this change is made; we suggest changes to both.)

1/T23) 4.12.2.2: END SEGMENT is not allowed in the metafile descriptor, only in global and local segments.

1/T24) Annex H: Only the mapping of GKS functions to CGM is given. Reference is made to Annex E for the reverse mapping, but such is not provided in Annex E. A reverse mapping should be given. The mapping should be from CGM element to GKSM item, rather than GKS function, since (for example) the correspondence between the two-vector CHARACTER ORIENTATION of CGM and the one-vector "character up vector" of GKS is not clear.

Technical/Editorial Comments

1/TE1) GLOBAL: Readability would be enhanced by defining pseudo-data types such as "CO: CI or CD, depending on COLOUR SELECTION MODE", and then using type CO for the SET <X> REPRESENTATION elements. (Similarly with line width, edge width, and marker size parameters.) Reference: CGI, and also CGM Part 3.

1/TE2) Global comment: All or almost all occurrences of the word "function" should be changed to "element".

1/TE3) Global comment: The words "appear" and "appearance" are overloaded by being used both for the visual appearance of a segment or primitive on the display and for the occurrence or presence of an element or segment in the metafile. The latter terminology is preferred.

1/TE4) Global comment: The phrasing throughout about "<primitive> is drawn" needs to be modified to take global segments into account, where it is actually not drawn until referenced in a picture.

1/TE5) page 2: METAFILE CATEGORY was omitted from the list.

1/TE6) The state diagram should be updated to reflect the new states: for readability, this might require more than one diagram. Although it can be argued that Table 3(a) contains all relevant information, the diagram greatly aids comprehension.

1/TE7) We suggest renaming category CGMEXT1 to CGMADD1, since this is officially an Addendum rather than an Extension.

1/TE8) 4.5.6, paragraph 3: "Another device viewport specification mode" should be DEVICE VIEWPORT MAPPING (see markup for other editorial changes in this paragraph).

1/TE9) The concepts section on Closed Figures (4.6.8) needs to use much more of the text from CGI: it is currently incomprehensible to those who don't already know it. At least one example with accompanying picture is needed as well.

1/TE10) 4.6.9: Use "spatial mapping" to distinguish this from mapping of colour spaces or mappings of elements from GKS to CGM.

1/TE11) A section should be added to clause 4 describing the difference between "pictures" (delimited by BEGIN PICTURE and END PICTURE) and "frames" (delimited by instances of PREPARE VIEW SURFACE). We suggest also adding a "note" (not part of the standard) which offers guidance on how to use a WISS and the metafile output workstation's open/active state to capture just the last frame in an interactive session.

1/TE12) 4.12.3.2 and elsewhere: All references to the "CGI pipeline" should be reworded. This document should be reworded to avoid discussion of a pipeline.

1/TE13) page 33: UPDATE has no default and should be deleted from the list.

1/TE14) Table 3(a): It should be clarified that many elements allowed in state PO can also occur in the METAFILE DEFAULTS REPLACEMENT, but are correctly listed as not being permitted in state MD. Perhaps another column should be considered for this table, for state DR (defaults replacement).

1/TE15) Table 3(a), formatting: Please repeat the column headings for each page, on Table 3(a). The spacing between columns is not consistent.

1/TE16) 5.3.17: The description needs to be rewritten: it should say what the MAXIMUM VDC EXTENT is. How it is used by GKS should be specified in Annex E and/or Annex H.

1/TE17) 5.5.9 says "see DEVICE VIEWPORT for a fuller explanation" -but 5.5.7 has no explanation (it should). Please include all of CGI's 5.3.5, including the discussion of mirroring, so that the element has enough information specified.

1/TE18) 5.5.10: The parameters bnig and bnil should be condensed to bni, to match the description and resolved issue.

1/TE19) 5.5.12: CGI resolved at Valbonne that the flag is applicable to both hardcopy and softcopy. We suggest calling the formal parameter "force clear", with enumerated values (forced, conditional).

1/TE20) 5.7.38: The parameters are not in same order as CGI: they should be.

1/TE21) 5.7.42: "All output operations" should become "rendering of primitives".

1/TE22) page 35, point 4: Although a user cannot specify skewed character up vector and character base vector at the GKS API, skewed vectors can result at the GKS WSI if the segmentation transformation is applied before writing the information into the metafile.

1/TE23) page 1: The objective of "graphical session capture" does not, on the surface, appear to justify the inclusion of elements such as

PICK IDENTIFIER and PICK PRIORITY. If the intent is to provide the ability to replay from CGM the beginning of an interrupted session in GKS and then continue from that point, this should be made explicit. Graphical session capture should be defined in the glossary.

1/TE24) Page 3, 4.4.4: This line should instead state what the element specifies; how GKS uses it should appear in Annex E or Annex H in a full discussion of how the mapping takes place.

1/TE25) Page 16: What is "Figure Closed" state? This doesn't match CGI and makes little sense. It should probably be "Figure Open/Region Closed".

1/TE26) Page 13: The document would benefit from an overall description of the states and their transitions, in addition to Table 3(a).

1/TE27) 5.5.14: MODIFY FONT LIST - second parameter should be nS (list of strings) rather than S.

1/TE28) 5.5.16: "ACTIVE" state should be replaced by "the state of the metafile prior to the most recent BEGIN FIGURE."

1/TE29) 5.5.17: 1st sentence, add "except in global segments, where rendering is deferred until the segment is referenced."

1/TE30) In the description of Closed Figures, we should probably not capitalize line and fill area functions: it would also help to state at the start that these refer to classes of primitives rather than specific ones.

1/TE31) 5.5.19, last sentence, "picture": Add "or global segment".

1/TE32) 5.5.20, notes 1 and 2: Add "for use in restoring the value: however, the current mode is not restored." Also, "escape function identifier" should be "escape element identifier".

1/TE33) page 17, note for page 51 of CGM: This should be clause 5.3.13, not 5.3.3.

1/TE34) page 16: The data type "name" appears never to be used: if this is so, it should be deleted.

1/TE35) page 30, 5.10.8 - The description is clear in the context of CGI, but needs to be rewritten to be clear in the context of CGM.

1/TE36) 4.5.4, second paragraph, line 1: "ready to accept graphics" would be clearer as "ready to accept graphical output".

1/TE37) 4.6.8: "curves" should be "arcs"

1/TE38) Page 4, 1st sentence: These appear to be examples of "softcopy devices: hardcopy devices are typically printers, plotters, and film recorders.

Draft U.S. Response to CGM DPAD 1 Part 1

Following are the U.S. comments on segmentation, Part 1.

Technical Comments

1/T25) Page 6, section 4.12.1, list of segment elements:

REOPEN SEGMENT should be a delimiter,
REDRAW SEGMENT is absent,
IMPLICIT REGENERATION MODE is missing the word "SEGMENT",
PICK IDENTIFIER is absent.

Add a paragraph following the list along the lines of: "BEGIN SEGMENT, REOPEN SEGMENT, and END SEGMENT are delimiter elements rather than segment elements, as they may not occur within SEGMENT OPEN state.".

1/T26) page 28, section 5.10.2, first paragraph:
Replace the first sentence with: "The contents of the identified segment are copied into the picture, in its current state." This is due to the corresponding changes in CGI.

1/T27) page 29:
Add a new section for REDRAW SEGMENT, which was omitted. Renumber the following sections appropriately.

"Parameters:
segment identifier, (SN)

Description:
This element is intended to result in a redraw of the identified existing segment, unless the segment's visibility attribute is INVISIBLE. The resulting display is implementation-dependent. Any segment overlapping the identified segment may also be redrawn, depending on the implementation.

The effect of this element is intended to be independent of the implicit segment regeneration mode."

Technical Editorial Comments

1/TE39) page 1, last paragraph, add:
Segments may be appended using REOPEN SEGMENT and END SEGMENT.

1/TE40) page 6, section 4.12.1, insert:
Throughout section 4.12 and its subsections, the concepts involving segments are described in the context of an interpreter's handling thereof. The CGM simply stores the elements for an interpreter's use. Interpretation of elements that do not affect final picture appearance is implementation-dependent, e.g. PICK IDENTIFIER. This implementation dependency may be determined by other Standards (e.g. GKS, PHIGS, etc.), as used by the implementation of an interpreter.

1/TE41) page 6, section 4.12.1, original first paragraph:
Rephrase "Segments may be:" to "Segments may have elements associated with them to control interpretation thereof:".

Rephrase list items a-f to be non-verbs, e.g.: transformation, visibility or invisibility, highlighting, front to back ordering, detectability or undetectability, and deleting.

Following the list, change "within a picture. They can..." to read "Segments may..."

1/TE42) page 6, section 4.12.2:
The phrase "Local Segments ... are local to that picture." is a circular definition. Rephrase to "Local Segments ... have no existence beyond the bounds of that picture."

1/TE43) page 6, section 4.12.2.1:
Add "Global Segments may not be reopened." following the first sentence. Replace the third sentence ("... not defined or known to ..." with "They are not a part of any picture within the metafile."). In 4.12.2.1, first line, "normal" should be deleted: a phrase along the lines of "which are the same elements used to define local segments" should be added. Also, "COPY" should be "COPY SEGMENT".

1/TE44) page 7, section 4.12.2.2:
END SEGMENT is not allowed in MD state. It is only allowed in SO state. State that BEGIN SEGMENT changes the state to GSD.

REDRAW SEGMENT is missing from the list of segment control elements.

REOPEN SEGMENT should not be a segment control element, but rather a delimiter like BEGIN SEGMENT.

Change the last statement of the second paragraph to a parenthetical comment on the first statement of that paragraph. e.g., "(see Table 3(a), ...)."

1/TE45) page 7, section 4.12.2.3:
Add the parenthetical comment "(see Table 3(a), ...)" to the end of the first sentence.

1/TE46) page 7, section 4.12.2.4:
The statement "every element that is modally defined and bound to primitives ... has a well defined value." is not clear. Reword to state that there are "defaults".

1/TE47) page 7, sections 4.12.3.1 through 4.12.3.3:
These sections should be 4.12.3.2 and 4.12.3.3, respectively, and the present section 4.12.3.3 should section 4.12.3.1, since ...3 defines terminology used by ...1 and ...2.

1/TE48) page 7, present section 4.12.3.2:
Rephrase "passing along the CGI pipeline are stored in" to read "are added to".

1/TE49) page 9, section 4.12.3.4:
Reword along the lines of:
"Non-retained data are those elements present in a picture, but not within local segments. Any handling of non-retained data by

an interpreter is implementation-dependent, beyond the requirement that simple display of a metafile picture must display that non-retained data."

1/TE50) page 11, section 4.12.5.2, first paragraph:
Change "prevent the waste of" to "permit an interpreter to save".

1/TE51) page 11, section 4.12.5.2, third paragraph:
Remove the first sentence, as it is inappropriate to a metafile specification.

1/TE52) page 12, section 4.12.6, first paragraph:
Change "open segment" to "picture definition in its present state".

1/TE53) page 16, section 5.2.6, Description, first sentence:
Rephrase to read: "This element delimits the start of a segment."

1/TE54) page 27, section 5.7.41, first paragraph:
Add: "Usage of the pick identifier is interpreter-dependent."

1/TE55) page 27, section 5.7.41:
Remove the second paragraph as it is inappropriate for a metafile specification.

1/TE56) page 28, section 5.10.1:
Remove the third paragraph as it is inappropriate for a metafile specification.

1/TE57) page 28, section 5.10.2, fourth paragraph:
Add to the last sentence: ", or whether the current state list attributes are applied."

1/TE58) page 28, section 5.10.2:
Remove the third paragraph as it is inappropriate for a metafile specification.

1/TE59) page 28, section 5.10.3:
Remove "NOTE -" and make a single paragraph. Change "BEGIN SEGMENT" to include "or RENAME SEGMENT".

1/TE60) page 29, section 5.10.4:
Delete the second through last paragraphs as they are inappropriate for a metafile specification.

1/TE61) page 29, section 5.10.5, Description:
Add: "Subsequent references to the segment must use the new segment identifier. The old segment identifier is freed for use in a subsequent BEGIN SEGMENT or RENAME SEGMENT."

1/TE62) page 29, section 5.10.7:
Delete the second paragraph through the last. Revise the second sentence of the first paragraph to read: "The IMPLICIT SEGMENT REGENERATION MODE element may occur in the PICTURE OPEN state only." Add a final sentence to the first paragraph: "The interpreter itself determines the actions required to support this element."

1/TE63) page 31, section 5.10.9:
The specification for the 'transformation matrix' parameter differs from that for COPY SEGMENT, and from the corresponding CGI specification.

1/TE64) page 32, section 5.10.10:
Delete the "NOTE" as it is inappropriate for a metafile specification.

1/TE65) page 32, section 5.10.13:
Add to the end of the first sentence: ", depending on the functionality of the metafile interpreter."

1/TE66) page 33, section 5.10.14:
Delete the second paragraph as it is inappropriate for a metafile specification.

The following comments concern the formal grammar defined for CGM DPAD 1. Where relevant, the errors are traced back to the CGM standard.

1/TE67) Non-recursive metafile contents production.

As written, the non-terminal <metafile contents> production (page 36) defines a production for a single picture preceded by 0 or more escape elements and/or 0 or more external elements, and followed by 0 or more escape elements and/or 0 or more external elements. According to the grammar specified in addendum 1, a single picture is the only permissible sequence. This is clearly in conflict with part 4 of the standard (specifically section 4.1, page 9) which defines the minimally correct metafile as:

```
BEGIN METAFILE
METAFILE VERSION
METAFILE ELEMENT LIST
END METAFILE
```

Whereas, according to the addendum 1 formal grammar, the minimally correct metafile would be:

```
BEGIN METAFILE
METAFILE VERSION
METAFILE ELEMENT LIST
<picture>
END METAFILE
```

The source of this inconsistency is the non-terminal production <metafile>, which is defined as:

```
<metafile>        ::= <BEGIN METAFILE>
                         <identifier>
                      <metafile descriptor>
                      <metafile contents>
                      <END METAFILE>
```

According to the CGM standard, the <metafile contents> production should be an optional production, as follows:

```
<metafile>        ::= <BEGIN METAFILE>
                         <identifier>
                      <metafile descriptor>
                      <metafile contents>*
                      <END METAFILE>
```

indicating that the <metafile contents> production can occur 0 or more times in the metafile between the <metafile descriptor> and the <END METAFILE> component.

The suggested changes have been highlighted.

1/TE68) <metafile contents> production changed from the CGM standard.

The production <metafile contents> has been altered from that which is defined in the current CGM standard. Currently, <metafile contents> is defined as:

```
<metafile contents>      ::=  <extra element>*
                            | <picture>
                            | <extra element>*
```

Addendum 1 defines the <metafile contents> production as:

```
<metafile contents>      ::= <extra element>*
                             <picture>
                             <extra element>*
```

According to the CGM standard, the variation of <metafile contents> defined in addendum 1 is correct and the variation in the standard is incorrect.

1/TE69) Inconsistencies between formal grammar and CGM standard/addendum.

The following list of inconsistencies between the formal grammar and the CGM standard or the CGM addendum #1 text were found:

1. Section 4.12.2.2 specifically allows for the GLOBAL SEGMENT state to be entered from the METAFILE DESCRIPTOR state, but the grammar does not reflect this. Due to the fact that Global Segments cannot accept the same class of elements as Local Segments, the definitions of the <picture element> and <optional descr elmt> productions will have to be modified, and the two new productions will be defined (<local segment element> and <global segment element>).

2. Section 4.3 of the standard specifically allows intervening escape or external elements between the <BEGIN METAFILE> and <metafile descriptor> components. According to CGM addendum 1 formal grammar (and also the CGM standard's formal grammar), no escape or external elements can occur between <BEGIN METAFILE> and <metafile descriptor>.

3. The formal grammar indicates an ordering of elements within the Metafile Descriptor State that is not defined by the CGM standard or addendum 1. The specific problems (i.e. differences between the grammar and the text of addendum 1) within the Metafile Descriptor State are listed below. In each case, the statement made is the interpretation of the formal grammar (addendum 1 and in some cases the CGM standard as well) and is not a restriction defined by the text of either addendum 1 or the standard:

   a. <METAFILE VERSION> must be the second element in a syntactically correct metafile.

   b. <METAFILE DESCRIPTION> can only occur after <METAFILE VERSION>.

c. ‹METAFILE CATEGORY› can only occur after ‹METAFILE DESCRIPTION›.

d. ‹METAFILE ELEMENT LIST› (which is a required element for a syntacticly correct metafile according to the standard) must occur before any of the optional metafile descriptor elements.

e. ‹METAFILE CATEGORY› is a dependent production associated with ‹METAFILE DESCRIPTION›.

4. The REOPEN SEGMENT element is in reality a delimiter element, and as such alters the production for local segments. A transition from the PICTURE OPEN state to the LOCAL SEGMENT state can occur on either a BEGIN SEGMENT element or a REOPEN SEGMENT element. This change is reflected in the new production ‹local segment element›.

Based on the above items, the following productions must be modified or added for the addendum 1 formal grammar to conform to the text of the CGM standard and addendum 1:

```
‹metafile descriptor›    ::=  ‹optional descr elmt›*
                              ‹element list›
                              ‹optional descr elmt›*
                              ‹version›
                              ‹optional descr elmt›*
                            | ‹optional descr elmt›*
                              ‹version›
                              ‹optional descr elmt›*
                              ‹element list›
                              ‹optional descr elmt›*

‹optional descr elmt› ::=  ‹VDC TYPE›
                              ‹vdc type›
                            | ‹MAXIMUM COLOUR INDEX›
                              ‹colour index›
                            | ‹COLOUR VALUE EXTENT›
                              ‹red green blue› (2)
                            | ‹METAFILE DEFAULTS REPLACEMENT›
                              ‹element default›+
                            | ‹FONT LIST›
                              ‹font name›+
                            | ‹CHARACTER SET LIST›
                              ‹character set definition›+
                            | ‹CHARACTER CODING ANNOUNCER›
                              ‹coding technique enumerated›
                            | ‹scalar precision›
                            | ‹escape element›
                            | ‹external element›
                            | ‹MAXIMUM VDC EXTENT›
                              ‹point› (2)
                            | ‹SEGMENT PRIORITY EXTENT›
                              ‹minimum extent›
                              ‹maximum extent›
                            | ‹METAFILE CATEGORY›
                              ‹category enumerated›
                            | ‹global segment element›
```

```
<version>                  ::= <METAFILE VERSION>
                               <integer>

<picture element>          ::=  <control element>
                              | <graphical element>
                              | <attribute element>
                              | <escape element>
                              | <external element>
                              | <local segment element>

<local segment element> ::=  <BEGIN SEGMENT>
                                <segment name>
                                <eligible local seg picture elem>
                             <END SEGMENT>
                             | <REOPEN SEGMENT>
                                <segment name>
                                <eligible local seg picture elem>
                             <END SEGMENT>

<global segment element>::=  <BEGIN SEGMENT>
                                <segment name>
                             <eligible global seg picture elem>
                             <END SEGMENT>
```

The suggested changes have been highlighted.

The above grammar will accomplish the following modifications to the interpretation of a given metafile:

1. The GLOBAL SEGMENT DEFINITION state can be entered from the METAFILE DESCRIPTOR state without allowing the elements specifically prohibited from occurring in the GLOBAL SEGMENT DEFINITION state from appearing. Transition from the METAFILE DESCRIPTOR state to the GLOBAL SEGMENT DEFINITION will occur on a <BEGIN SEGMENT> element.

2. The LOCAL SEGMENT DEFINITION state can be entered from the PICTURE OPEN state without the restrictions that apply only to global segments. Transition from the PICTURE OPEN state to the LOCAL SEGMENT DEFINITION state can occur on either BEGIN SEGMENT or REOPEN SEGMENT.

3. The modified <metafile descriptor> production requires both <METAFILE VERSION> and <METAFILE ELEMENT LIST> to appear in the Metafile Descriptor state, but does not imply any specific ordering of any of the Metafile Descriptor elements, including the two required Metafile Descriptor elements. The production also allows for external or escape elements to appear before any Metafile Descriptor element. However, we recomend that the addendum text be modified to specify the positions of <METAFILE VERSION>, <METAFILE ELEMENT LIST>, <METAFILE DESCRIPTION> and <METAFILE CATEGORY>. This, of course, could mean that the above grammar is also be incorrect. To

implement the US proposal, the ‹metafile descriptor› and ‹optional descr elmnt› productions would have to be changed to:

```
‹metafile descriptor›   ::=  ‹extra element›*
                             ‹METAFILE VERSION›
                             ‹METAFILE ELEMENT LIST›
                             ‹METAFILE CATEGORY›o
                             ‹METAFILE DESCRIPTION›o
                             ‹optional descr elmt›*

‹optional descr elmt›   ::=  ‹VDC TYPE›
                               ‹vdc type›
                           ¦ ‹MAXIMUM COLOUR INDEX›
                               ‹colour index›
                           ¦ ‹COLOUR VALUE EXTENT›
                               ‹red green blue› (2)
                           ¦ ‹METAFILE DEFAULTS REPLACEMENT›
                               ‹element default›+
                           ¦ ‹FONT LIST›
                               ‹font name›+
                           ¦ ‹CHARACTER SET LIST›
                               ‹character set definition›+
                           ¦ ‹CHARACTER CODING ANNOUNCER›
                               ‹coding technique enumerated›
                           ¦ ‹scalar precision›
                           ¦ ‹escape element›
                           ¦ ‹external element›
                           ¦ ‹MAXIMUM VDC EXTENT›
                               ‹point› (2)
                           ¦ ‹SEGMENT PRIORITY EXTENT›
                               ‹minimum extent›
                               ‹maximum extent›
                           ¦ ‹global segment element›
```

4. The modified ‹optional descr elmnt› production allows the transition to the GLOBAL SEGMENT DEFINITION state from the METAFILE DESCRIPTOR state, and also removes the restrictions on the METAFILE CATEGORY element.

The non-terminal productions ‹identification›, ‹metafile category›, ‹metafile description›, ‹characteristics›, ‹segment›, and ‹eligible picture element› productions have been deleted.

Some of these errors also occur in the current CGM standard's formal grammar. The above productions must be modified to correct the current formal grammar.

1/TE70) Undefined non-terminal production ‹attribute element›.

The non-terminal production ‹element default› (page 38 of CGM addendum 1) references a non-terminal production ‹attribute element›, yet nowhere in addendum 1 or the CGM standard is there defined a non-terminal production ‹attribute element›. It appears that this is a simple editorial problem, and that what was intended was ‹primitive attribute element›. Either the reference

to ‹attribute element› must be changed to ‹primitive attribute element› or the non-terminal production ‹primitive attribute element› (page 43 of CGM addendum 1) should be changed to ‹attribute element›.

This error occurs in the CGM standard as well as CGM addendum 1.

1/TE71) Multiple definition of non-terminal production ‹point›.

The non-terminal production ‹point› is defined identically in both sections F.3.2. page 39 (Metafile Descriptor Elements) and F.3.3. page 39 (Picture Descriptor Elements). In keeping with the convention used in the rest of the formal grammar, a production should only be defined in the section which first uses the production (in this case, section F.3.2).

This error does not occur in the CGM standard.

1/TE72) Violations of the CGM standard in PARTIAL TEXT state.

The PARTIAL TEXT state is entered by the following production:

```
‹nonfinal character list›   ::= ‹NOT FINAL›
                                ‹string›
                                ‹character attribute list›
                                ‹spanned text›
```

There are multiple problems with this production and the productions which are invoked as a result. The following problems have been noted:

1. The non-terminal production ‹character attribute list› is not defined in the grammar. A production ‹char attribute list› is defined which appear to be what was intended. This is just an editorial problem which does not exist in the CGM standard.

2. The production ‹char attribute list› does not accurately reflect the elements which are permitted to occur in the PARTIAL TEXT state. To reflect the syntax specified by the standard, a new non-terminal production ‹partial text attributes› should be created and the ‹nonfinal character list› production modified as follows:

```
‹nonfinal character list› ::= ‹NOT FINAL›
                              ‹string›
                              ‹partial text attributes›*
                              ‹spanned text›

‹partial text attributes› ::=  ‹TEXT FONT INDEX›
                                  ‹positive integer›
                               |‹TEXT PRECISION›
                                  ‹text precision enumerated›
                               |‹CHARACTER EXPANSION FACTOR›
                                  ‹real›
                               |‹CHARACTER SPACING›
                                  ‹real›
                               |‹TEXT COLOUR›
```

```
                                    <colour>
                                |<CHARACTER HEIGHT>
                                    <non-negative vdc value>
                                |<CHARACTER SET INDEX>
                                    <positive index>
                                |<ALTERNATE CHARACTER SET INDEX>
                                    <positive index>
                                |<TEXT BUNDLE INDEX>
                                    <positive index>
                                |<AUXILIARY COLOUR>
                                    <colour>
                                |<TRANSPARENCY>
                                    <on-off indicator enumerator>
```

The suggested changes have been highlighted.

The elements which are permitted to occur in the PARTIAL TEXT state are defined in the CGM standard by both Figure 12 (page 40, State diagram) and section 5.6.6 (page 64, APPEND TEXT).

This error occurs in the CGM standard as well as CGM DPAD 1.

1/TE73) Editorial inconsistency in <TEXT REPRESENTATION> production.

The comments for the character expansion factor and the character spacing components of the <TEXT REPRESENTATION> alternative of the <representation element> production (page 46) are associated with the wrong components based on the description of TEXT REPRESENTATION in section 5.7.38 (page 25). The production reads

```
<representation mode> ::=        .
                                 .
                                 .

                          | <TEXT REPRESENTATION>
                             <positive index>
                             <index> (font)
                             <text precision enumerated>
                             <real> (character spacing)
                             <real> (expansion factor)
                             <colour>
                                 .
                                 .
                                 .
```

yet the text of the addendum indicates the production should read as

```
<representation mode>  ::=       .
                                 .
                                 .
                          | <TEXT REPRESENTATION>
                             <positive index>
                             <index> (font)
                             <text precision enumerated>
```

```
                                    <real> (expansion factor)
                                    <real> (character spacing)
                                    <colour>
                                       .
                                       .
                                       .
```

The suggested changes have been highlighted.

1/TE74) Editorial inconsistency in <FILL REPRESENTATION> production.

The fill colour specifier component of the <FILL REPRESENTATION> alternative of the <representation element> production (section F.3.6, page 46) is in a conflicting position with respect to the definition of FILL REPRESENTATION (section 5.7.39, page 25) in the addendum 1 text. The production currently reads

```
<representation mode>  ::=      .
                                .
                                .
                             | <FILL REPRESENTATION>
                                 <positive index>
                                 <interior style enumerated>
                                 <index>  (hatch index)
                                 <positive index> (pattern index)
                                 <colour>
                                  .
                                  .
                                  .
```

According to the description of FILL REPRESENTATION in section 5.7.39, the production should read:

```
<representation mode>  ::=      .
                                .
                                .
                             | <FILL REPRESENTATION>
                                 <positive index>
                                 <interior style enumerated>
                                 <colour>
                                 <index>  (hatch index)
                                 <positive index> (pattern index)
                                  .
                                  .
                                  .
```

The suggested changes have been highlighted.

1/TE75) <SET DEFERRAL MODE> production substituted for <DEFERRAL MODE>

The non-terminal productions <control element> (section F.3.4, page 39), <eligible control element> (section F.3.2, page 38) and <element name enumerated> (section F.4, page 51) all reference a terminal production <SET DEFERRAL MODE> which is never defined. It appears that the intended terminal production was <DEFERRAL MODE>. The element in the text of the addendum is DEFERRAL MODE.

1/TE76) <metric scale factor> mistyped.

The <metric scale factor> production (section F.3.4, page 40) was mistyped as:

<metric scale factor> ! <real>

which is a semantically null production, as well as incorrect syntax for a BNF grammar. The correct BNF for this production is the same as was specified in the CGM standard:

<metric scale factor> ::= <real>

The suggested modification has been highlighted.

1/TE77) Complete the productions for those elements not yet defined.

The productions <eligible local seg picture elem> and <eligible global seg picture elem> must be completed from the state table.

U.S. Response to CGM DPAD 1 Part 2

The U.S. disapproves of Part 2 of CGM PDAD 1 (SC 24 N 20) with the following comments:

Technical Editorial Comments

2/TE1) Page 1, "Add the following to table 1:"

Add the 8 bit opcodes.

2/TE2) Page 2, 8.2.16

GLOBAL: Add a blank line before the production of a non-terminal as the CGM-IS text does. For example:

```
<METAFILE-CATEGORY-opcode:  3/1 3/0>
<enumerated: metafile-category>
**** blank line goes here ****
<enumerated: metafile-category> = <integer: 0> (cgm)
                                  <integer: 1> (gksm)
                                  <integer: 2> (cgmaddl)
```

Use "cgmaddl" instead of "cgmextl".

2/TE3) Page 2, 8.2.17

<point: first corner> and <point: second corner> should be called <point: high value> and <point: low value> as in the functional specification (5.3.17).

2/TE4) Page 3, 8.4.8

"NOTE: Default not consistent with GKS" is not appropriate here. Perhaps it should be in Clause 6.

2/TE5) Page 3, 8.4.10

"SET DEFERRAL MODE" should be called "DEFERRAL MODE" as in the functional specification.

2/TE6) Page 3, 8.4.14

<font-declaration> is inconsistant with the functional specification and the other encodings. Replace <font-declaration> with <font-names>.

2/TE7) Page 4, 8.4.20

The data type of attribute set name is defined as ASN in the functional specification, but is shown as integer in the character encoding. The character encoding should be <asn: attribute-set-name> instead of <integer: attribute-set-name>.

2/TE8) Page 4/5, 8.5.21

The last four parameters are not in the functional specification or in other encodings and should be removed.

The order of the parameters is inconsistant with the functional specification (5.7.39).

3/TE10) Page 6, 7.10, Table 11

INHERITANCE FILTER default should be: n/a,1.

3/TE11) Page 6, 7.10, Table 11, note 2

"P7: (real) y translation component" should be "P7: (vdc) y translation component".

3/TE12) Page 6, 7.10, Table 11, note 8

Parameter numbers are missing.

3/TE13) Page 7, "The following forms sub-clause 7.11:"

See GLOBAL note at the beginning of the comments on the binary encoding as to whether there should be two classes.

3/TE14) Page 8, 7.11, Table 12, note 2

The order of "visible" and "invisible" is reversed from the EDGE VISIBILITY element in the CGM-IS text.

3/TE15) Page 8, 7.11, Table 12, note 6

The last sentence incorrectly uses the phrase "...segment display priority..." instead of "...segment pick priority..."

U.S. Response to CGM DPAD 1

The U.S. disapproves of Part 4 of CGM PDAD 1 (SC 24 N 22) with the following comments:

Technical Comments

4/T1) Page 8, 6.11

As per the notes in previous encodings, combine 6.10 and 6.11 into one group of elements.

Technical Editorial Comments

4/TE1) Page 1, "Add the following to the end of sub-clause 5.3.5:"

The description of TM incorrectly calls the matrix a "real" transformation when it includes VDC types as well as reals.

4/TE1) Page 1, "Add the following to the end of sub-clause 5.4.3:"

Suggested abbreviation changes: DEV instead of DEVICE, VPORT instead of VIEW.

The U.S. recommends that the Language Bindings Working Group be consulted to insure consistency of abbreviations used in this document.

4/TE3) Page 1/2, "Add the following to the end of sub-clause 5.4.4:"

BACK is the abbreviation for BACKGROUND in the CGM-IS text. The U.S. suggests that BKWD be used as the abbreviation for BACKWARDS. The U.S. also suggests using HILITE instead of HIGHLIGHT as the abbreviation for HIGHLIGHTING. The U.S. suggests not abbreviating RESTORE.

4/TE4) Page 2, "Add the following to the end of sub-clause 5.4.5:"

CIRCULAR ARC CENTRE BACKWARDS ARCCTRBACK should be consistent with whatever is chosen for BACKWARDS in sub-clause 5.4.4. Suggested ARCCTRBKWD.

4/TE5) Page 3, "Add the following to the end of sub-clause 6.3:"

Replace CGMEXT1 with CGMADD1.

HIGH and LOW is used in the functional specification and should be used here instead of FIRST and SECOND.

4/TE6) Page 3, "Add the following to the end of sub-clause 6.5:"

The note about defaults does not belong here.

4/TE7) Page 3/4, "Add the following to the end of sub-clause 6.5:". DEFERRAL MODE

DEFERRAL MODE has no parameters so it does not need <SOFTSEP>.

4/TE8) Page 3/4, "Add the following to the end of sub-clause 6.5:", MODIFY FONT LIST

There should be <OPTSEP> instead of <SEP> before <S:FONTNAME>.

<SEP><S:FONTNAME> should be <<SEP> <S:FONTNAME>>.

4/TE9) Page 3/4, "Add the following to the end of sub-clause 6.5:", CHARSETDESIGNATOR

First <SEP> should be <SOFTSEP> as in the original encoding. The <HARDSEP> should be replaced with <SEP> as in the original encoding.

4/TE10) Page 5, "Add the following to sub-clause 6.6:", PIXEL ARRAY

Remove <LOCLCOLRSPEC>.

4/TE11) Page 5/6, "Add the following to sub-clause 6.7:", TEXT REPRESENTATION

The order of <SPACING> and <FACTOR> is reversed from the functional specification.

4/TE12) Page 5/6, "Add the following to sub-clause 6.7:", FILL REPRESENTATION

Parameter order does not match functional specifications.

4/TE13) Page 8, 6.11 , SEGMENT VISIBILITY

Reverse the order of VIS!INVIS

APPENDIX 4

SUMMARY ARTICLE OF THE BLAKENEY MEETING OF THE SPECIAL WORKING GROUP ON FUTURE PLANNING

Graphics Standards

The Future of ISO Graphics Standards

George S. Carson,
GSC Associates

Janet Chin of Chin Associates, a regular contributor to this column, asked that George Carson, who chaired the meeting discussed here, contribute his knowledge to the column. "I think that the CG community should know about this important work as soon as possible," said Chin. We thank George Carson for his time and effort in bringing us up to date.

At its first plenary meeting in Berlin during December of 1987, ISO/IEC JTC1/SC24 (the computer graphics committee) recognized the need to evaluate the work undertaken by its predecessor committee (SC21/WG2) and to plan for the future development of graphics standards. SC24 authorized the formation of a Special Working Group on Future Planning with the charter to recommend a five-year strategic plan for organizing the work of SC24. The group was also to address the feasibility of developing a unified structure for relating computer graphics standards, to identify new areas of computer graphics that should be covered by a standard, and to make appropriate statements regarding the relationships of computer graphics standards with standards in other areas. In addition the group was chartered to define areas in which computer graphics standards need to be reevaluated or extended or where new standards need to be developed to meet user requirements.

The Special Working Group met at Blakeney, England, from April 11 to April 13, 1988. Sixteen delegates from six national bodies were in attendance. The group was able to reach consensus regarding how graphics standards should be developed in the future and how the present work of SC24 should be concluded. This consensus was presented to SC24 at its plenary meeting held earlier this month in Tucson, Arizona.

Lessons learned

The group spent some time enumerating lessons learned since the formation of SC21/WG2. These lessons are

1. There should be fewer implementation dependencies in SC24 standards.
2. Precise scope and goals must be agreed to before development starts on a New Work Item (NWI).
3. Allowing enhancements to occur during the standardization process thwarts timely standards.
4. Continuity of staffing cannot be assumed throughout a standards activity.
5. Timely development is required if standards are to be useful.
6. A single reference model would have expedited current standards activities significantly.
7. The text model in the current generation of computer graphics standards is inadequate.
8. The input model adopted for GKS, GKS-3D, and PHIGS is inadequate to specify quality user interfaces.

Trends

The group reached consensus that the following trends exist and should influence future work of SC24:

1. Windowing technology cannot be ignored.
2. Integration with standards in other areas has become essential.
3. Domain specific standards are becoming more necessary.
4. There is increasing use of graphics.
5. De facto "standards" have begun to influence computer graphics standards work.

Options for progressing SC24 work

The Special Working Group defined and considered seven options for progressing SC24 work:

1. Produce a basic reference model by the 1989 SC24 meeting and suspend all SC24 work not at the DIS (draft international standard) stage by that meeting.
2. Start work on a new framework for describing computer graphics standards and use this model to restructure all work not at the DIS stage by the 1989 meeting.
3. Improve existing procedures based on lessons learned.
4. Suspend all work until a new reference model for computer graphics standards is developed and reaches the DIS stage.
5. Go on in the usual way and disregard all lessons learned from the past.
6. Abandon graphics standardization development.
7. Acknowledge, evaluate, and recognize de facto standards.

In evaluating these seven options, the group considered a number of criteria. These included

- the impact of adoption of the option on the timeliness of computer graphics standards
- the quality of standards produced
- the possible increase or decrease in staffing of Rapporteur Groups
- the risk involved to the eventual production of a family of high-quality computer graphics standards

The Special Working Group concluded that options 1 and 2 represented the most realistic ways to proceed. Option 3 was considered a rather poor fallback option. The group recommended that a strategy

0272-1716/88/0700-0082$01.00 © 1988 IEEE

be devised based on an appropriate combination of options 1 and 2 that would synthesize their best features.

Component/framework model for the development of standards

The Special Working Group concluded unanimously that a reference model is needed to unite SC24 standards, and that standards should be developed in a different manner in the future from that used in the past. The group recommended an approach to developing standards which they called the "Component/Framework Process." In this approach *components* might be, for example, data types, operations, or attributes, and *frameworks* might be considered the "glue" that takes a set of data types and operations and constructs a standard from them. The group recognized that considerable technical work is needed to decide the best way of structuring the SC24 standards in terms of components and frameworks and to define necessary formal methodologies and interface techniques to allow portions of standards to be developed independently as components and subsequently integrated by frameworks.

The group suggested the following items that might be components in SC24 standards:

- primitive data types
- rendering attribute data types
- transformation operations
- storage data types

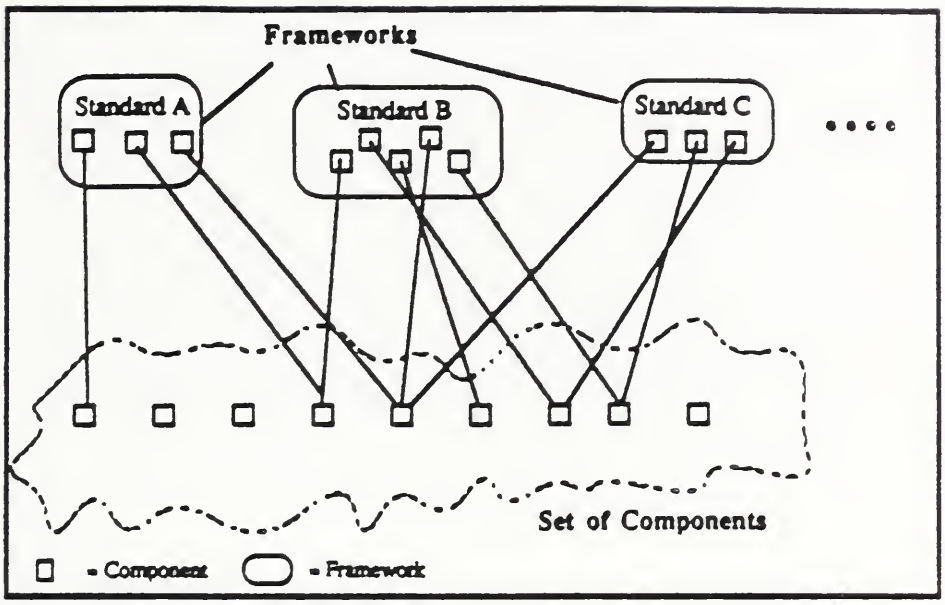
A framework might include

- data types
- sequencing of operations
- deferred modes
- display timing and control

The figure illustrates how components relate to frameworks in the development of standards. Sets of components are tied together into one or more standards by a framework structure. Compatibility is achieved among various standards, since they incorporate common components.

Recommendations

The Special Working Group made a number of recommendations to SC24 and its working groups. Five output documents were produced for consideration. These output documents include a description of the component framework model for developing standards and a five-year planning document for SC24. The principal recommendations of the Special Working Group were the following:



Components and Frameworks.

1. SC24 should adopt a new structure for the next generation of computer graphics standards, which will require a new development process.
2. SC24 should suspend or restructure under the new development process any projects for semantic standards not at the level of DIS or DAD (draft addendum) by the 1989 SC24 meeting. (The effect of this recommendation is to divide SC24 standards into a first generation of semantic standards, including GKS, GKS-3D, PHIGS, CGM, and—assuming work can be completed in time—CGI and CGM addenda, and a second generation of semantic standards developed under the new component framework process.)
3. SC24 should consider windowing and imaging as part of the next generation of graphics standards.
4. SC24 standards must interwork with standards in the areas of office systems, image capture, and Standards for the Exchange Product Model Data (STEP). SC24 standards must also function in an Open Systems Interconnection (OSI) environment and as components of an Open Distributed Processing (ODP) environment.

Conclusions

If the recommendations of the Special Working Group are endorsed by all of SC24, they will have a profound impact on the future of computer graphics standards. They should force work already under way, which has been severely delayed by "creeping features," either to be completed within the next year or to be dropped. SC24 will have the opportunity to develop a second generation of standards based on an integrated reference model, thereby avoiding many of the "compatibility" problems which have been perceived in the first generation of computer graphics standards. ■

George S. Carson is president of GSC Associates, a systems engineering consulting firm specializing in real-time signal and information processing systems. He has more than 15 years experience building systems incorporating multiple processors, computer networks, and interactive graphical user interfaces. Carson was vice convenor and chief liaison officer for ISO TC97/SC21/WG2, where he was also the Rapporteur for Formal Description. He is the convenor of the ISO/IEC JTC1/SC24 Special Working Group on Future Planning and serves as document editor for SC24 reference model work. In ANSI X3H3 he is chair of X3H3.2, with responsibility for work on formal descriptions, reference models, user requirements, and X3H3 external liaisons.

Carson received a PhD in mathematics from the University of California at Riverside.

This page left intentionally blank.

APPENDIX 5

FIVE-YEAR PLAN PRODUCED BY SWG/FP

This page left intentionally blank.

I S O ISO/IEC JTC1/SC24 N ____

ISO

International Standardization Organization
Organization Internationale de Normalisation

ISO/IEC JTC1/SC24

Computer Graphics

Secretariat: FR of Germany (DIN)

Title: Towards a Five Year Plan for SC24

Source (Country, Organization, Committee): Special Working Group on Future Planning

Status:
- ☐ SC24 Output Document
- ☐ Member Body position
- ☐ Expert opinion (ALL expert contributions submitted to SC meeting must be authorized by the Member Body).
- ☒ Working Group output

Type:
- ☒ A1: "Immediate" Output Document from WG/SC24 Meetings (to arrive at SC24 Secretariat 2 weeks after meeting).
- ☐ A2: "Delayed" Output Document from WG/SC24 Meetings (to arrive at SC24 Secretariat 4 weeks after meeting)
- ☐ B: National/Expert Contributions to WG/SC24 Meetings (to arrive at the WG or SC Secretariat 8 weeks before the beginning of the the WG or SC meeting)
- ☐ C: Recent Liaison Documents (to arrive at the expert by the start of the WG or SC meeting)

Type of Distribution:
- ☒ Through SC24 Secretariat
- ☐ Direct to Member Bodies

Date of Submission: 14 April 1988

Keywords: Future planning

1. Introduction

This document describes deliberations of the SC24 Special Working Group (SWG) on Future Planning, which met at Blakeney, UK on 11-13 April 1988. This group was chartered by SC24 to make recommendations on how best to proceed with the development of Computer Graphics standards. The formal recommendations of the SWG are contained in a separate document (SC24 N_____). This document describes how the group assessed the current state of Computer Graphics, including lessons learned from past standardisation efforts and future trends. It describes seven alternative strategies defined to represent the range of options available to SC24 in the pursuit of its future work. The strategies are evaluated and a course of action is recommended. Finally, the likely effects of the group's recommendations on the five year plan of work of SC24 are assessed.

2. Current state

2.1 Introduction

The Special Working Group spent some time considering the current state of Computer Graphics standardisation, reviewing the style of working and looking at the possibilities of new work. The aim was to see what, if any, consensus could be achieved and how that would effect the way future work was organised.

To get the largest breadth of ideas, the group split into five subgroups and the conclusions of the subgroups were consolidated into this portion of the report. Each sub-group was asked to consider future trends, predict what effect such trends would make on future work and whether this would lead to new areas for standardisation. Each subgroup was asked to enumerate the lessons learned from the current activities. In particular, they were asked to assess the relative advantages of standardising current practice and developing standards appropriate to perceived future requirements.

2.2 Lessons learned

1) There should be fewer implementation dependencies in SC24 standards

The usefullness of existing standards has been impacted by the high level of implementation dependencies. This has caused problems in portability and made conformance difficult to assess. A strong recommendation was that future standards should have much fewer implementation dependencies.

Implementation dependencies should not be a preferred way of achieving consensus in controversial areas. There was a realisation that some of these implementation dependencies were introduced to allow support in a wide range of (sometimes obsolete) devices, and that support for such devices might be more difficult under the new guidelines.

2) Precise scope and goals must be agreed to before development starts on a NWI

Much effort has been wasted in the past arguing over the relationships among standards. These arguments could have been avoided by spending more time getting precise definitions and agreement of scope and goals initially. Relationships, or the lack thereof among standards must be explicitly agreed before starting development.

3) Allowing enhancements to occur during the standardisation process retards timely standards

All current standardisation activities have sufferred from creeping enhancements that appeared

during the review. Adding one more attribute, primitive, or whatever impacts the scope and goals, requires the agreed functionality to be reevaluated and may violate the original reference model. It was believed that this had severely impacted the timescales for current standards. If such enhancements are anticipated as necessary due to outside development, they must be part of the planning process and scheduled in.

4) Continuity of staffing cannot be assumed throughout a standards activity

It takes at least four years to produce a standard. With the current movement of graphics staff it is unreasonable to believe that the same team will remain intact throughout a standards development. Consequently, there is a need to capture arguments and decisions in a precise way so that education of members is not achieved by reworking old resolved arguments.

5) Timely development is required if standards are to be useful

The large elapsed time for standardisation has meant that features are often out-of-date by the time the standard is complete. This causes pressure for de facto standards to be ratified without lengthy evaluation. One approach to solving the problem would be to concentrate on encapsulating current practice which could be standardised more quickly. The alternative is to make a new standard progressive in all areas so that it is still timely after it appears.

Existing standards have tended to be a mixture of current practice and progressive features and concepts. This means that standardisation takes a long time and some features are out of date when the standard appears.

6) A single reference model would have expedited current standard activities by a significant amount

A major cause of long timescales has been the lack of a Reference Model into which the various standards fit. A framework for new activities must be agreed in some detail or the same problem will occur in the next generation of standards.

7) Problems with text

It has become increasingly clear that the text model adopted by the current generation of computer graphics standards is inadequate, and is also incompatible with the work done within SC18. Thus, as SC24 is not the correct place to develop a new text model, future graphics standards should integrate the work done on text within SC18.

8) Deficiencies with graphics input

The Input Model adopted by GKS, GKS-3D, and PHIGS is inadequate to specify quality user intefaces, and development of a more powerful model is required.

2.2 Trends

There was consensus on the following points:

1) Windowing technology cannot be ignored

The current standards are already being used in a windowing environment. The trend is for computer graphics to be used mainly in windowing environments in the future. The windowing system will make demands on the graphics system and will constrain what is allowed in the graphics system.

There is an urgent need for some windowing facilities to be included in the current generation of standards. The next generation of standards must include windowing management.

2) Integration with standards in other areas has become essential

The first generation of standardisation assumed that computer graphics was responsible for all the output to a device and did not need to interface with other standards. Today, graphics is being used in a distributed environment with multiple windows and shared devices. Furthermore, we have seen the increasing need for documents and displays involving multiple media, such as integrated text and graphics. This means that computer graphics has to be more aware of other standards and be able to coexist with them.

3) Domain specific standards are becoming more necessary

The use of computer graphics is expanding into new areas. This has meant that it is less likely that one standard will be appropriate across all application areas to be possible. Users will demand what is obtainable.

4) Increase in the use of graphics

The use of computer graphics has become pervasive, with graphics incorporated in almost all application areas. The resulting increase in user population will make standards more important in the future.

5) Influence of de facto "standards" on our work

The trend towards the use of workstations in a heterogeneous distributed environment has increased the drive towards de facto "standards" available on a range of equipment. The pressure to formalise such useful standards will increase as the user population wishes to access such systems and wishes to insure that implementations conform to such "standards."

3. Possible ways forward

To find out how best to proceed with the SC24 work, several alternate potential processes were developed. These options were then judged by the following four criteria:

- time required
- quality of results
- staffing required
- risks

3.1 The seven options considered

Option 1 - BSI option, see SC24 N 111

- produce a Basic Reference Model by the 1989 SC24 meeting;
- all work not at DIS stage by the 1989 SC24 meeting will be suspended;
- completion of the CGM addenda and CGI by the 1989 meeting should be encouraged;
- no new project will progress to DP before the Basic Reference Model is developed;
  a) accept no new work items at all;
  b) accept and work on new work items in parallel;
- as part of their revision cycle, standards will be reviewed in light of the Basic Reference Model
- work on the Basic Reference Model could be viewed as the first step in, and prerequisite for, the GKS revision processs;

Option 2 - see output document from this meeting entitled "Component Process for Development of Standards"

- start work on Component/Framework (CFW) project;
- all work not at DIS stage by the 1989 SC24 meeting will be restuctured using the CFW model and/or a new generation of standards could be produced;
- as part of their revision cycle, standards will be reviewed in light of the Basic Reference Model;
- kernel components exist, but not necessarily a kernel framework;
- NWI could be accepted and would be related to the CFW model

Option 3 - Improve existing procedures based on lessons learned
- focus on the GKS review and CGI completion;
- Alternatives for Reference Model work:
  a) no new Reference Model work;
  b) address Reference Model issues during reviews;
- continue CGM addenda work;
- process NWI's as usual;

Option 4 - Suspend all work until Reference Model DIS
- suspend all work not now at DIS stage;
- concentrate on the Reference Model work;
- approve no New Work Items until Reference Model DIS;
- after having developed a complete Reference Model, define a second generation of graphics standards.

Option 5 - Go on in the usual way and disregard all the lessons learned from the past.

Option 6 - Abandon graphics standardisation development

3.2 Evaluation of the options

The SWG split into three subgroups each of which judged two of the options (it was assumed that Option 6 could be assessed without further discussion). Besides the few criteria given above (time required, quality of results, staffing required, and risks) several other considerations influenced the evaluation of the options:

- possible loss of people/effort
- possible loss of credibility
- need to have a Basic Reference Model or a Component Framework
- need to interwork with other standards

Table 3.1 gives an overall assessment of each option against the major criteria.

	CRITERIA			
Options	Time	Quality	Staffing	Risks
1	6	7	7	8
2	8	7	7	5
3	6	5	5	8
4	2	8	10-2	3
5	5	4	4	2
6	-	-	10	-
7	8	2	9	3

Table 3-1 Assessment of options

Having discussed and assessed the seven options, the SWG believed that options 1 and 2 were the most realistic ways to proceed, with option 3 as a rather poor fallback option. It was believed that some strategy based on a combination of 1 and 2 might produce a superior option to all seven examined. Therefore, the SWG urges SC24 to adopt an approach for future work based on a synthesis of the best features from these two options.

Comments on the options rejected were:

Option 3 (Improve Existing Procedures)

The major advantages to this approach would be continuity, the ability to respond to NWI quickly, and production of compatible standards.

The major disadvantages were :

1) Experience has shown that the current method of generating standards is a very long process.
2) It is difficult to get effort into Reference Model Work while it is seen as a peripheral activity.
3) Work in the Reference Model area will tend to be located with the Work Items leading to incompatible reference models between standards.
4) It is unlikely that the current approach will produce a set of standards that easily fit into a distributed or rendering environment.
5) JTC1 has requested work on a Reference Model. This approach is unlikely to satisfy them.
6) The GKS Review is unlikely to attract CGM and PHIGS experts.
7) It will not put any deadlines on CGI and the work is likely to extend.

Option 4 (Stop all work until complete Reference Model produced)

The advantage of this approach was the possibility of producing better quality results than other options. However, there was some concern that producing a reference model in a vacuum with no parallel practical work might detract from the quality, resulting in no better quality than Options 1 or 2.

The major advantages were:

1) It is believed that the SC would lose about a third of the existing staff and that it would be difficult to get them back to standards work after the moratorium.
2) It would have a negative effect on both countries and constituencies which would impact the credibility of the SC.
3) By insisting on a full Reference Model before work would restart, would cause at least a 2 year delay.
4) An over-rapid development of a complete Reference Model might produce a flawed product.
5) Work would continue on de facto standards inside ISO and in other SCs. This would have an adverse effect when work is restarted.

Option 5 (Business as usual)

The SWG was very concerned with applying past techniques to future work. Current SC24 methods are close to breakdown and not capable of meeting future needs.

In the GKS Review, either minor corrections would be made or a new version based on a revised GKS Reference Model. Using current procedures, it is impossible to judge which of those approaches would be adopted.

Additional disadvantages:

1) Much effort is wasted due to the fragmentation of the work and the continual need to re-assess each work item due to lack of future planning and early consensus on the goals. Consequently, schedules are very uncertain.
2) Results are poor due to lack of integration and incompatability of concepts. Consequently leverage in the marketplace is poor.
3) Current process strains relations between delegations due to incompatability of objectives.

Option 7 (Rubber stamp de facto standards)

This option does have several advantages:

Towards a Five Year Plan

1) Standards are likely to be produced relatively quickly (less than 2 years).
2) Staffing may well be easier.
3) Standards will already have been implemented with a defined user base. Consequently, acceptance will be high.

The disadvantages which override the advantages are:

1) Long term acceptance of standards is jeopardized. Industry tends to produce new systems relatively quickly. Life of individual standards will be short.
2) The quality of staff working on standards will decrease. It will tend to attract staff with a narrow perspective.
3) Compatability of standards is abandoned.
4) Standards will tend to overlap.
5) Incompatabilities are likely to arise between the de facto standard and its ISO equivalent due to the different speeds of revision.
6) Difficult to respond to users' requirements.
7) Stability of user environment will suffer, leading to increased costs.
8) Consensus may be difficult to achieve. Politics may play a greater part. There could even be legal problems.
9) The current set of standards would be abandoned.

4. Recommended 5 year plan

4.1 Projects likely to be affected by 1989 "cutoff"

Under the two options for significant procedural changes proposed by the SWG on Future Planning, mid - 1989 forms a milestone in the processing of SC24 projects. Those projects which are not at DIS or DAD stage would either be suspended or restructured. It is anticipated that the following semantic definition projects will be advanced enough and will not be affected:

CGI
CGM Addendum 1
(PHIGS)
(GKS-3D)

The following existing and new proposed projects will likely be affected:

CGM Addendum 2
CGM Addendum 3
PHIGS +
Windows
Imaging
Reviews (GKS, CGM, PHIGS, CGI, ...)

Basic Reference model work will receive increased emphasis under both options, and will not be affected by the deadline.

The SWG recommends that there should be no effect on LAB and Validation and Testing work. There are two options for Registration:
- no effect;
- new proposals are not accepted for some time.

4.2 Structure of Work prior to the 1989 SC24 meeting.

Both options for procedural changes affect the structure of work for new and uncompleted projects. The implications for the affected projects listed above are given in the next two sections. In both cases, completion of the CGI and CGM Addenda is encouraged. It is felt that the CGI and CGM Addendum 1 will have progressed far enough (DIS / DAD) before the cutoff date.

Option 1: Up to and after the 1989 SC24 meeting, work on a Basic Reference Model (BRM) proceeds.

1) Work not at DIS / DAD: experts working on projects are urged to estimate carefully the chances of reaching the DIS stage before the cutoff date.

2) Review: review of GKS, CGM, PHIGS, CGI, etc. will commence when they are considered for review.

3) NWI: CGM Addendum 3, PHIGS +, Windowing, Imaging; two options are identified:
   a) work allowed, but not beyond WD, until BRM finished;
   b) no work until BRM finished.

The majority felt that the first was better - as it would allow the development of NWI work, limited to WD stage.

Option 2: work on a Basic Reference Model - defining component types and frameworks - proceeds up to and after the 1989 SC24 meeting.

1) work not at DIS / DAD: experts working on projects are urged to estimate carefully the chances of reaching the DIS stage before the cutoff date.

2) reviews: review of GKS, CGM, PHIGS, CGI, ...; will comence when they are considered for review. The effect of the new components/framework (CFW) process is not yet determined.

3) NWI: CGM Addendum 3, PHIGS +, Windowing, Imaging; could proceed under the new CFW; this would probably mean some delays while the components of the CFW were advanced sufficiently and the NWI were given frameworks.

4.3 Structuring work after 1989

With the adoption of either option 1 or option 2, we must consider how the work of SC24 would be organised after the 1989 meeting. There are two obvious alternative ways to proceed. Under the first option, work would continue to be organised according to the first generation standards where possible, with additional rapporteur groups added to accomodate the new work areas.

Under the second option, work would be re-organised according to a component / framework decomposition. First generation standard revision would be carried out by "mapping" these standards into the new framework.

Figure 4.1 illustrates a likely progression of work under option 1. Review of each standard would start approximately four years after it became an IS. During this review, each standard would be restructured into a framework in accordance with the new Basic Reference Model. Reference Model work to develop component models would continue after 1989. [CGI review is not illustrated since it would begin beyond the five year period considered here].

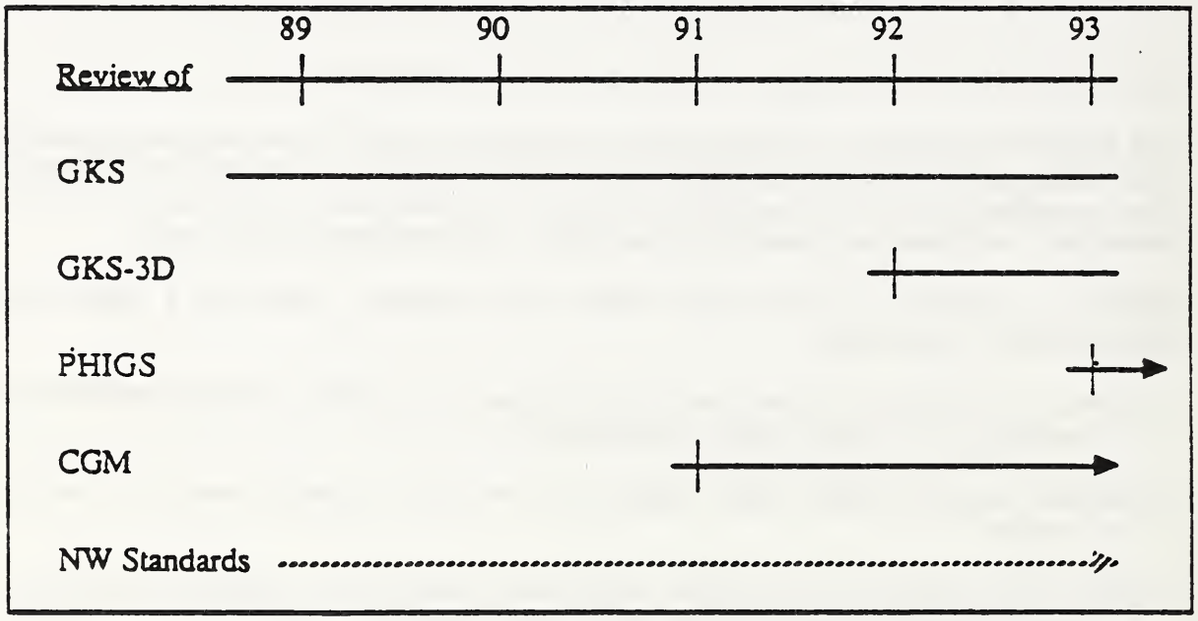


Figure 4-1 Progression organised according to Current Work

Towards a Five Year Plan

Figure 4.2 illustrates one possible progression of SC24 work under the second option. Reference Model work would continue developing at least the core components. Some components -- such as those involved with imaging or windowing -- might be developed by separate rapporteur groups, but sufficient interesting work must remain within the Reference Model Rapporteur Group to attract qualified experts to the work. New work would be developed as appropriate.

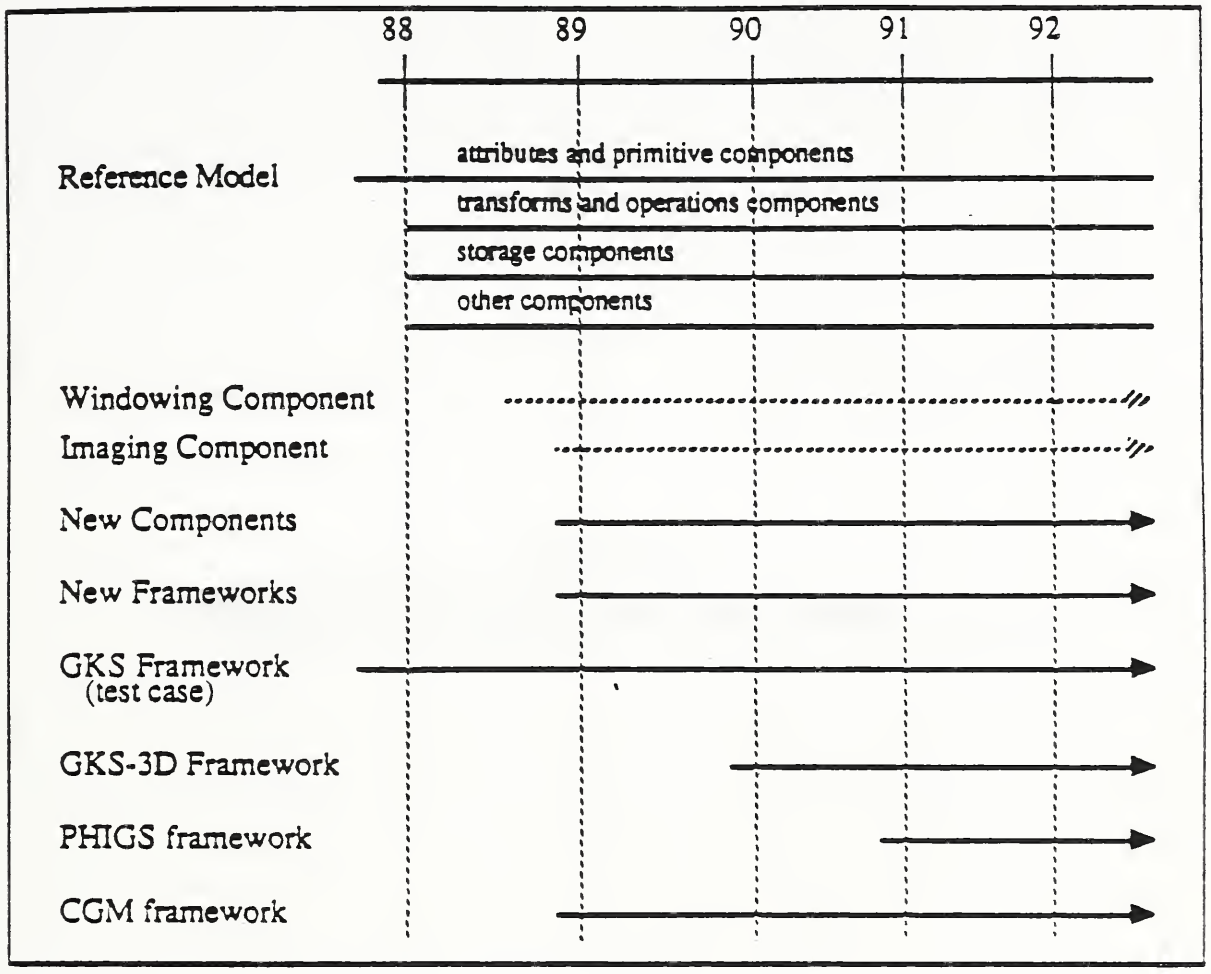


Figure 4-2 Progression organized according to Component/Framework Model

4.4 Extensibility

Future SC24 standards must consider requirements for user-defined extensibility.

4.5 Management of work

Management of work would be at the SC24 level.

This page left intentionally blank.

APPENDIX 6

DOCUMENTS FROM BLAKENEY METAFILE RG MEETING

'P' National Bodies SC 24	Approve	Approve with Comments	Disapprove	Abstain
Austria		X		
Belgium				
Brazil	X			
Czechoslovakia			X	
Denmark				
France			X	
Germany, F. R.			X	
Hungary				
Italy		X		
Japan		X		
Netherlands				
Sweden				
Switzerland				
United Kingdom			X	
United States			X	
USSR	X			

Minutes of ISO/IEC/JTCI/SC24/WG3 CGEM Meeting

18th April 1988

Blakeney Hotel, Norfolk, England

Present

E Moeller, L Henderson, W Brandenburg, A Francis, A Mumford, D Arnold (part of day)

1. Welcome

E Moeller thanked D Arnold on behalf of the group for organising the meeting in such pleasant surroundings.

2. Report of Rapporteur

E Moeller thanked A Mumford for preparing Addendum 1 for the SC24 meeting.

Addendum 2 was produced later than had been hoped and SC24 have not put this out for ballot.

D Arnold to handle WG3 mailing.

3. Report on Special Working Group Meeting

D. Arnold reported on this meeting. Recommendation 2 from the meeting proposed that SC24 should have DIS or DAD text ready for ballot by the end of the SC24 1989 meeting. If not the work will be suspended or restructured.

Addendum 2 was discussed in this light. There was concern that the document may not meet this deadline.

Addendum 3 was not discussed in detail. The need for the work was discussed implicitly.

The SWG had discussed the way that future work might progress. The work may be split up into groups which discuss components of standards.

4. Tucson meeting - strategy for addendum processing - frequency of meetings

The meeting at Tucson for CGM work is 4 days. It is hoped that editing work and CGI liaison may continue till the Tuesday of the second week.

The frequency of meetings was discussed. The balance between the different projects, the amount of work, good representation causes problems.

An interim meeting on all 3 addenda might attract good attendance. A week long meeting possibly in early 1989 maybe in the U.S.

Possible schedule - tightest

mid September	document out
end September	distributed
end December	comments in
end Feb/start March	editing meeting

This is the tightest schedule possible. The advantages and disadvantages of producing a draft out of an interim meeting were discussed. A March meeting seems favourite.

5. Discussion of Comments to Addendum 1

(i) General

Relationship CGI/CGM

It is a major problem that the group do not have the new CGI document. It was agreed that we would have to accept the comments regarding the current proposals e.g. Austrian comments 7 (clarification) for this meeting.

(a) *Austrian Comment*
Austrian comment re UPDATE - do we need to explain the MAKE PICTURE CURRENT/UPDATE differences. A clarificatic note may be required. An alternative is to make UPDATE just UPDATE 'perform'.

Agreed it is desirable to remove redundancy. Alternatives:

a. UPDATE only allowed for 'perform'
b. remove MAKE PICTURE CURRENT

There was discussion on CGI liaison. There is need for liaison at the CGEM meeting at Tucson. This is mainly for technical assistance. It is hoped that a number of delegations could have experts available.

It was agreed we wished to be close to CGI and that we would have an element for update with 'perform'. CGI has access to the flag 'new frame necessary at update'.

UPDATE (POSTPONE) maps to MAKE PICTURE CURRENT

UPDATE(PERFORM) - it is proposed that a new element PERFORM DEFERRED ACTIONS is included in Addendum 1 to replace the current UPDATE.

In IMPLICIT SEGMENT REGENERATION MODE need to describe the actions which can be deferred (page 29).

A paper is to be written on the update issue.

(b) REGION OPEN/CLOSE to be removed as in CGI.

(c) Use inheritance filter for REOPEN - in CGI this only applies to COPY. Agreed we should do same as CGI. Believe that effect can be achieved with SAVE and RESTORE attributes.

Discussion on COPY. DOES COPY need to be included in the metafile? Is it just for Global Segments. It depends where the CGM is in the CGI pipeline. Metafiles would be larger if COPY is not there. Agreed that COPY should be retained.

CSSR Comments

CGI has to be technically stable now if it is to meet the deadline (assuming SC24 accepts SWG recommendations).

Should the 2 addendum be merged? Straw vote - 3 NO 2 ABSTENTIONS.

Technical comment 4 - description on segments needed.

Technical comment 8 - REDRAW SEGMENT is missing - to be included.

UK Comments

These were discussed in full while A Francis was present.

1. Re MODIFY FONT/CHARACTER SET LIST. These were included at Valbonne because there is no start index for the lists. There was concern that a high numbered font might be used in GKS and might cause a sparse font list.

   It was suggested that the GKS font number might have a high value but that the registered name would be included as the next font in the font list.

   The use of the modify prevents the generator writing the descriptor last.

   Straw vote

   3 in favour of removing the elements
   2 against

2. SEGMENT PRIORITY

   Integer or real - agreed that should follow CGI. Whatever decision that is.

3. Is CGI Part 6 stable enough for inclusion of PIXEL ARRAY and DRAWING MODE?

   If still within CGI then we will leave then in.

   US comment suggests adding local colour precision this was agreed.

   Should there be a compressed encoding for PIXEL ARRAY?

4. Segment elements should be split to segment control and attributes. At Valbonne they were joined to

allow just one class in the binary encoding. It was agreed to separate these.

5. DRAWING MODE - should use enumerated types - follow CGI.

6. UPDATE - discussed.

7. DEVICE V'PORT SPEC UNITS - should have same default as GKS - follow CGI.

8. Elements in Figure Open State - CGI allows more flexibility than CGM regarding the appearance of events in states such as text open.

   Agreed that we should maintain the CGM philosophy and make Figure Open Similar to Text Open.

   Allow the fill attributes as well.

9. SAVE PRIMITIVE ATTRIBUTES saves all attributes but INHERITANCE FILTER allows selectivity. Needs to be consistent. Need to check current CGI text.

Metafile Categories and Element Sets

It is necessary to complete the following table:

Element Set	Category		
	cgm	cgmadd1	gksm
DRAWING_SET	X		
DRAWING_PLUS_CONTROL	X		
GKSMO			X

France have suggested a category for static structured pictures.

There is an individual US position suggesting a static picture plus global segments and the new primitives.

The US have suggested that there should be an element set for GKSM (as well as GKSMO). Should the shorthand have a different name?

I was agreed that there was no need for a GKSMO category. It was also agreed that the GKSMO element list would remain. France have suggested 5 categories. It was felt that 4 were possible via category or shorthand now. The static image is the current CGM plus the element list. It was felt it could be extended for structures as global segments but just used as a symbol library. i.e. A static picture plus structures which cannot be handled dynamically plus other additional elements. The French proposal suggests that this structured image could also be modified using bundle table modifications.

The French proposal for the structured image would be in the GKSM category. The static structured picture would be in the CGMADD1 category. Both these may be useful as symbol libraries.

There was uncertainty as to why some of the other elements are omitted. It is necessary to have some more clarification.

It was agreed that few categories and more element lists were required.

All categories therefore include structures and the subsets are element lists which conform to a category.

Defer decision on Tucson recommendation until Tuesday.

France

France note CLIP INDICATOR and CLIP RECTANGLE have been made a single function. This is incompatible with CGM - raise with CGI group.

Pixel array is in Part 6 of CGI which implies a different encoding class from output primitives as used in Addendum 1 for this element.

France ask for clarification of relationship between GKS and CGM. The mapping will be included. It was agreed that the precisions do not need to be passed to the application layer.

Germany

T1-2 to T1-4 - agreed to move DP, change description of VSP and add VSU to be closer to CGI.

United States

Discussion (based on U.S. T8) on the rigorous structure to a metafile. There has been a move of VDC EXTENT. This comment suggests the majority of Picture Descriptor elements should be allowed in other states. This is not justified under the scope and goals of the addendum work.

Page 1-7 T25
REOPEN should be a delimiter element - agree.

Page 1-7 T27
REDRAW ALL SEGS - missing - agree

T26 - agree change

T10 - agree, as BSI comment addressed earlier

T11 - agree

T12 - agree - error in text

T13 - agree - add a note to the table indicating Metafile Closed State can exist with BEGIN METAFILE being the only element allowed.

T19 - document should follow "null value rule" e.g. invisible, visible - agreed this should be done, also ensure that CGI is the same.

T20 - as T19

T21 - agree - modify table.

T22 - agreed - see earlier discussion.

TE52 - agreed - wording should be changed.

TE63 - transformation matrix for SEG TRANS is different in CGI - agreed to edit.

Japan T5 - copy segment as US comment - agree

USA/T1 General comment on errors - most of this is related to the CGI text copying for this draft.

USA/T9 - does the grammar imply an order which is not in the functional description. Ordering is sensible in the Metafile Descriptor. This is not the case in the CGM IS text.

Germany/T1 suggests the Met Elem List and Met Category should appear before first global segment.

Agree in principle but need to decide the way to do this.

Would version 2 metafiles which are of category 'cgm' (default) also have this implied category.

19th April 1988

Present

E Moeller, L Henderson, A Mumford, T Hewitt, W Brandenburg, D Arnold (part of day)

1. Addendum 1 - Continuation of Technical Work

Category and Element Sets

Austria

Comment A1/T1 refers to element sets and metafile categories - it was agreed that we now had a clearer definition and the text would be improved.

It was agreed we have 3 categories 'cgm', 'cgmaddl', 'gksm'.

This comment also relates to the discussion the day before on the French and a US individual position.

Do we wish to have further element sets? There was some sympathy with the idea of a structured static picture to allow the new primitives to be used together with global segments. This gives the ability to have a symbol library for a static picture.

It was agreed that clarification and further discussion at Tucson was required.

It was noted that making the category and element list orthongonal results in a large number of 'profiles'. Should we allow all possible combinations of element class and category?

It may be useful to indicate in the standard which 'profiles' are particularly useful.

CGI Related Issues

D Arnold clarified the situation of the CGI document. Document out April/May. Tucson will be first to pass over comments. Aug/Sept editing meeting to produce second DP which will be out by end of year.

Pixel array has been redefined as a bitblt operation. This may therefore not be relevant for Addendum 1 as it is no longer an output primitive.

Most of major issues have been addressed by the CGI group.

GKS Related Issues

A/T16 - request for description of mapping COPY/INSERT to be changed to be as GKS.

It was agreed that GKS COPY and INSERT should be mapped as Annex E and not be COPY SEGMENT.

Other Comments

United States

US/T5 Scaling Mode and Device Viewpoint. It was agreed that the text needs to be changed.

US/T7 RENAME and DELETE should not be in global segments. It was agreed at Valbonne that DELETE should not be allowed. RENAME was not listed in the Valbonne minutes. Agreed that RENAME does not cause problems, but the same is true for DELETE if the MD is progressed sequentially. There was no use seen for the elements in GS state. Agreed with Valbonne decision on DELETE and extend to RENAME as US/T7 comment.

T14 - agree - Annex D to be extended.

T15 - DEVICE VIEWPORT default should be (0,0) (1,1) not i.d. Need to check CGI.

T16 - PICK ID default - agree, as CGI

T17 - agreed to make edit. Should the grammar be part of the standard if we are defining conformance via the grammar.

This may be a problem for backwards compatibility with CGM version 1.

T18 - Annex E needs paragraph re generating correct CGM font information - agreed.

T23 - END SEGMENT is not allowed in MD - needs to be.

TE1 - no we cannot define new data types because of backwards compatibility. Maybe an amendment?

TE11 - request for text on the differences between pictures and 'frames'. It was agreed this would be useful. No support for adding information on capturing frames in WISS.

TE14 - new state Defaults Replacement (DR).

TE16 - (also Austria) need to rewrite text for MAXIMUM VDC EXTENT.

2. Parts 2-4

Mainly relate to Part 1 changes. Decided to leave this.

3. Amendments to CGM is Text

It was agreed that amendments would be useful. These are to be listed by Tucson and the procedure for formalising these sorted out by SC24.

4. Addendum 3

As a minimum we need to ensure that the Tucson meeting makes a decision on the work. It was suggested that one option might be for SC24 could recognise the value of the work. The components of the work could be working on in SC24.

It was agreed that there was interest in the work progressing. The SC24 deadline causes problems for the split of the work. The US has people interested in the work. There are problems for the metafile group in discussing new functionality for graphics standards. There is a need to discuss these elements across application interfaces and metafiles etc.

Group to recommend to SC24 that the work is interesting and relevant. Although the group is aware of the SWG recommendations the group feels that this work should be progressed in some form at SC24.

5. Addendum 2

Need to sort out the scope and purpose that can be achieved given the SWG recommendations on timescale.

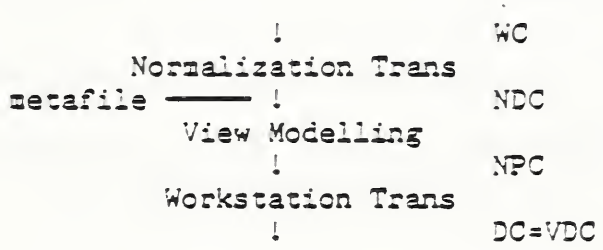
SC24 has approved GKS-3D extension. The London meeting found that it was straightforward to use global segments to extend the functionality towards PHIGS.

E Moeller noted the good quality of the document and thanked T Hewitt. Terry noted that the document was a team effort.

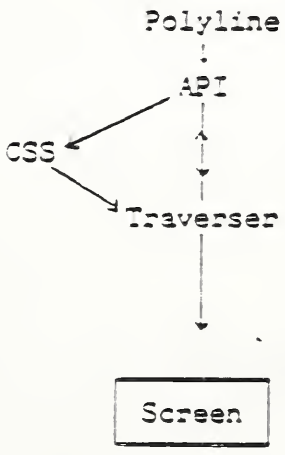
The timescale was such that no comments are available. There are therefore no national comments.

There was clarification of the technical reasons surrounding the current draft.

GKS-3D metafile is stored NDC



PHIGS



To access a workstation in PHIGS - open it and post structures. A metafile output workstation can be defined. If an element is replaced what is written on the metafile.

Functions are written to the archiver explicitly and this is therefore cleaner than a metafile.

If CGM is written below API then need to store EXECUTIVE STRUCTURE, if below traverser then this is a flattened structure. In Addendum 2 theses are categories PHIGS and PHIGSMO. The PHIGS category could be used as an archiver.

There are only a few elements need to go from PHIGSMO to PHIGS.

The metafile workstation has to handle continuous traversal.

PHIGS already has an archive file - is duplication a good idea. The PHIGS archive clear text former is not the same the clear text encoding of CGM.

There was some discussion on the merits and problems of mixing 2D and 3D coordinates in a metafile. All coordinates going to the workstation will be 3D. The PHIGS archiver has 2D and 3D coordinates because it goes out and comes in at the same point in the pipeline. This is not necessarily the case of CGM.

The archiver is not adequate for a hard copy capture mechanism. The metafile workstation is (probably) too flexible for this. Need to open, post and close.

The archive file format is based on the CGM. The need for a consistent line format across the API standards is important.

Should we be using the PHIGS archive file as the basis for work? Should there be character and clear text encoding for the archive file.

T Hewitt to write an explanation of what is in the current document.

The decisions of Egham and Valbonne to do at least GKS-3D support were reaffirmed. It was agreed that any functions now withdrawn from GKS-3D and PHIGS will be removed from Addendum 2.

6. <u>Allocation of Work</u>

(a)	Addendum 2 description, scope and goals	WTH
(b)	UPDATE clarification	LH
(c)	CGI questions	EM
(d)	Mapping CGM Elements to GKS Item Types	EM/AM
(e)	GKS Grammar	WB
(f)	CGM Amendments	LH/AM
(g)	Statement for SC24 on Addendum 3	LH
(h)	op-codes	
(i)	Technical changes to Addendum 2	

7. Op-codes

The BSI have produced a draft CGI character encoding. They have suggested some changes on the op-code tables in Addendum 1.

Agreed to use one class for segment functions and that a single class should also be used in the binary encoding.

SC2/WG8 - suggested PICK ID should be in the segment class rather than the attributes. The group felt that it was more appropriate to keep it with the primitive attribute elements.

It was agreed to move BEGIN FIGURE to 3/3 3/0 and to move the rest of the elements following that element.

The 'device viewpoint' data type referred to by SC2 has been addressed by the German technical comments.

OUTPUT DOCUMENTS

1.	Minutes	SC24/WG3/N15
2.	Addendum 2 clarification	SC24/WG3/N17
3.	Addendum 3 recommendation	SC24/WG3/N18
4.	UPDATE clarification	SC24/WG3/N16
5.	Draft disposition of comments	SC24/WG3/CGM
6.	Amendments to IS text	SC24/WG3/CGM
7.	GKSM Grammar	
8.	Mapping GKS Elements to item types	

ARCHIVES AND METAFILES - very rough draft

W.T. Hewitt

Introduction

This document will clarify the functionality supplied in CGM Addenda 2 for the support of metafiles and archivers. of PHIGS. The next sections describe the current status and intent of the various documents involved. Following is a discussion on the problems and a list of possible solutions.

Pertinent Documents

DIS 9592-1 PHIGS
DIS 9592-2 PHIGS ARCHIVER
DIS 9593-3 PHIGS ARCHIVER - Clear Text Encoding
IS 8632 CGM
ISO/IEC/SC24/N19 CGM Addenda 1 Functional Specification
ISO/IEC/SC24/N23 CGM Addenda 2 Functional Specification

PHIGS and Archiving

PHIGS has explicit support for Archives via functions such as OPEN ARCHIVE, ARCHIVE STRUCTURE NETWORKS etc, as specified in part 1. Parts 2 and 3 describe a file format for the archiver. The application snapshots the Central Structure Store with functions such as ARCHIVE STRUCTURE NETWORKS and a copy of the appropriate structures are despatched to the file. While this is good and satisfies a large user requirement NO workstation dependant information is stored, e.g. the view representations or bundle representations.
PHIGS Part 2 The Archive File Format states:
0.2 Reasons for this Standard
The main reasons for introducing a Standard PHIGS archive file are:
a) to allow structure definitions to be stored in an organised way on a graphical software system.
b) to facilitate transfer of structure definitions between different graphical software systems.
c) to enable structure definitions to be transferred between different computer graphic installations.
0.8 Relationship to othe Standards
This standard draws extensively for its model of an archive file format on ISO 8632....
It appears the intent is to support metafile type functionality! But if exchange between sites etc is required there should also be a character encoding and binary encodings.

PHIGS Metafile

PHIGS also has a metafile workstation facility. Again to extract from PHIGS DIS 9592
4.9 PHIGS Metafile Interface
For the purpose of long-term filing of graphical information, PHIGS provides an interface to file called graphical metafiles. They can be used for:
a) transporting graphical information between systems:

b) transporting graphical information from one place to another (for example by means of magnetic tapes);
c) transporting graphical information from one graphical application to another (for example, between PHIGS applications and applications using other graphical standards);
d) storing accompanying non-graphical information..

The major difference to the archive above is that workstation dependent information IS stored in the metafile. The standard does NOT explicitly exclude a metafile which can support the structure mechanism. However because this interface is a workstation if a structure is being edited that has been posted to a workstation of category MO many traversals will occur and the document gives no indication to what to do - whether to store ALL versions of that particular structure, or only the one that was effective at some particular time. Annex H (which is NOT part of the standard) discusses this. Further it is not clear from PHIGS where in the pipeline the metafile information leaves and re-enters. For example, what happens when an INTERPRET ITEM function is actioned to handle a SET VIEW REPRESENTATION? Does it go to all open workstations?

CGM, and CGM Addenda 1

Both of these documents have the same clause 0.2:
0.2 Reasons for this International Standard
The main reasons for producing a standard computer graphics metafile are:
a) to allow picture information to be stored in an organised way on a graphical software system;
b) to facilitate the transfer of picture information between different graphical systems;
c) to enable picture information to be transferred between graphical devices;
d) to enable picture information to be transferred between different computer graphics installations.
CGM was not sufficient to support GKS and hence the introduction of addenda 1

CGM Addenda 2

This addenda was added primarily to support GKS-3D. At that time it was very likely that namesets etc would be added to GKS-3D. The implication of that was taken into account at the London Meeting. It became clear at that time that by adding three functions to the CGM a great deal of support for PHIGS could be supplied. Those functions were EXECUTE STRUCTURE, SET LOCAL TRANSFORM and SET GLOBAL TRANSFORM. All the other functions necessary to support GKS-3D and PHIGS were already in the document. The addition of these three functions now enable the CGM A2 to serve as a support for:
a) The Workstation Interface of PHIGS (i.e. a devices which can support the structuring features of PHIGS) - session capture
b) The Archiver (no representation functions) - nearly picture capture
c) A dumb device driver for PHIGS i.e. a device which cannot support structures - picture capture.

Discussion

Clearly for a PHIGS implementer there is now the choice of three standards to help him develop his implementation:

a) CGM for the metafile Workstation
b) PHIGS Archiver (as in 9592 PArts 2 and 3)
c) CGM Addendum 2
The goals of the areas discussed above are so close that it is natural to bring them together.After all GKS-3D and PHIGS are supposed to be part of a family of compatable Graphics standards.

As to the interfaces. The Archive interface to PHIGS is good. The functionality is well defined and there are no ambiguities. (if the user wants several copies of the structure store, representing the status after each edit the user mus EXPLICITLY request those. Thus there will be no degradation of the interactive performance. The problem is that the user cannot get a hardcopy, i.e. an archive with the representations; except that the following piece of code is used:
open workstation(hardcopy device)
post structure network(fred,hardcopydevice)
close workstation(hardcopy device)
The disadvantage of this functionality is that if the user forgets the close workstation, then the situation is NOT defined clearly and unambiguously. I think the metafile workstation type should be removed from PHIGS and the functionality of the archiver be extended to allow the workstation dependent bits be added! For example the following instructions would be provided:
open archive
output structure network and representations to archive (call this XYX)
close archive
Conceptually there would be a workstation state list in PHIGS and the representation functions would set/inquire those entries. The information is only written to the archive when the function XYX is called. The action of XYX is:
1) NEW PICTURE
2) Output bundle reps
3) Output structures
4) end frame

That is XYX explicitly creates a frame in the archive!

Options

Follows are a list of options to rationalize the situation. There are extremes and the expectation is some more moderate solution is feasible

a) Drop PHIGS part 2 and part 3, use CGM addenda 2
b) Drop the PHIGS sections of CGM Addenda 2, Add a character and binary encodings to PHIGS Archiver
c) Upgrade CGM Addenda 2 to be a superset of CGM, CGM Addenda 1, CGM Addenda 2 (GKS-3D support) and PHIGS part 2 and part 3.
d) Remove the PHIGS workstation category MO and MI.

I do not believe the computer graphics community is served by several file formats which are almost compatible.

Proposed SC24/WG3 CGM Position on Addendum 3

The CGEM Rapporteur Group within SC24/WG3 has reviewed the proposed NWI for CGM Addendum 3. The group recognizes that the material contained therein is currently appearing in various defacto standards and is clearly related to the graphics standards activity of SC24. WG3 recommends that SC24 undertake work on the new graphics capabilities encompassed by the scope of Addendum 3. It is recognized that the new functionality is of interest not only to metafiles, but to API and other areas as well. WG3 therefore recommends that the group studying this new functionality be comprised of experts from all areas of SC24 - CGM, CGI, API standards, Reference Model, Registration, ...

APPENDIX 7

ADDENDUM 3 DRAFT FROM FAIRFAX MEETING

ANSI X3H3

Information Processing Systems --

Computer Graphics --

Metafile for the Storage and Transfer
of Picture Description Information

Scope Of Specification

Addendum 3

Draft Document 1.1

June 21, 1988

Purpose

The purpose of this addendum is to extend the CGM to effectively fulfill the picture transfer requirements of:

1) Engineering drawing and technical illustration
2) Graphics arts quality pictures, including geometric graphics, raster images, and text
3) Technical publishing

An additional intent is to keep pace with the graphics requirements of office systems, especially ODA requirements.

Scope

This addendum comprises a set of elements which will extend the capabilities of CGM as needed to meet additional user requirements in engineering drawing, graphics arts, and technical publishing. The set of elements should include all elements necessary to meet those requirements. It should be the minimal set sufficient to meet those requirements effectively.

The following preliminary list of capabilities is identified as necessary to meet these requirements.

1) Advanced 2D graphics, to include:
   - Bezier curves
   - Rational B-splines
   - Parametric spline curves
   - Line attributes of cap, miter, and join
   - Composite line primitive
   - User-defined line types
   - User-defined hatch styles
   - Arbitrary text path
   - Conics, and conic arcs

2) Text and font model of ISO 9541, Information Processing--Font and Character Information Interchange.

3) Picture composition and control, to include:
   - Arbitrary clipping boundary (general closed curve)
   - Shielding
   - Alignment

4) Additional color models beyond RGB
   - CIE
   - CMYB
   - Named colors

5) Additional raster graphics (scanned image) capabilities

6) Symbols: external reference to "standard" libraries of named symbols

The scope of this addendum assumes that the capabilities of CGM addendum 1 and addendum 2 are available.

Justification

CGM user have found that in some application areas the present standard provides a general framework that is suitable but lacks some functionality required by these applications. These areas include engineering drawing, the preperation of graphic arts quality presentation materials, and technical publishing.

A recent workshop sponsored jointly by the NBS and Eurographics, entitled "The CGM in the Real World", examined this issue and concluded that the CGM lacked capabilities to effectively meet some advanced user needs. As an example, for engineering drawingsm it is difficult if not impossible to effectively represent some higher-level constructs, e.g. splines and curves. Though such constructs can be simulated with simpler primitives ain the CGM at present, it is immposiblr to maintain accuracy and visual continuity and still retain device independence.

In all cases, there is a requirement to expand CGM text to include font definition capabilities that are consistent with the ISO DP9541 font standard. The font definition as it exists in ISO 8632 (CGM) is too general for practical use. Even though several fonts are identified in the TOP V3.0 CGM application Profile, these fonts are not adequate for publishing and graphics arts applications.

Many publishing and graphic arts systems use color models other than RGB. For efficiency and ease of implementation in these areas, for example, additional color models are needed, It also became apparent at the workshop that the Cell Array primitive in the CGM is not adequate for most applications that use raster graphics. Thus, this addendum will also provide additional raster graphics capabilities.

Advanced User Requirements for CGM

User Requirment	Future Need	(CALS) Eng Drw	Tech Man	Business Graphics	Office Systems	Publishing
Advanced 2-D Rqts.	Bezier Curves	X	X	X	X	X
	Path	X	X	X	X	X
	Pen	-	?	-	-	?
	Closed Figures	X	X	X	X	X
	User-defined linetypes	X	X	X	X	X
	Cap, join, miter	X	X	X	-	X
	B-Splines	X	-	-	-	-
Text / Fonts	ISO DP 9541	-	X	X	X	X
	Compatible Text and Fonts	-	X	X	X	X
	Arbitrary Path	-	X	X	X	X
Compound Documents	Arbitrary Clipping bdy	X	X	-	-	X
	Alignment	X	X	-	X	?
	Shielding	X	X	X	-	X
Color	Interpolated Fill	-	X	X	-	X
	CMYB	-	-	-	-	X
	CIE	-	-	-	X	X
	Name the colors	-	-	-	-	X

Advanced User Requirements for CGM (continued)

User Requirment	Future Need	(CALS) Eng Drw	Tech Man	Business Graphics	Office Systems	Publishing
Edita-bility	Application structures	X	X	X	X	X
	Segmentation	X	X	X	X	X
	Efficient Random Accesss	X	X	X	-	X
	Macros	X	X	X	X	X
Compact-ness	Storage	X	X	-	X	-
	Transfer	X	X	X	-	-
Image	Raster Attributes	X	X	X	X	X
	Device Independant Raster Data	X	X	X	X	X
Symbols	External Named Libraries	X	X	X	X	X
	User Defined Internal Libraries	X	X	X	X	X
Patterns/ Hatches	User Defined	X	X	X	X	X
	Additional Standardized (Carson)	X	X	Some	Some	X

ANSI X3H3

Information Processing Systems --

Computer Graphics --

Metafile for the Storage and Transfer

of Picture Description Information

Part 1

Functional Specification

(Clause 3)

Addendum 3

Draft Document 1.1

June 21, 1988

Subclause 3: Add or change the following entries:

colour model: A specification of a 3D colour coordinate system and a 3D subspace in the coordinate system within which each displayable colour is represented by a point. Some colour models include a fourth, redundant, dimension to allow the independent representation of black. For the purpose of ISO 8632 colour model refers to one of RGB, CIE L*u*v*, or CYMK.

colour selection mode: Indicator as to whether colour selection is to be direct (by specifying a colour value) or indexed (by specifying an index into a table of colour values). See COLOUR VALUE.

colour representation method: Indicator as to which of three colour models (RGB, CIE L*u*v*, CYMK) or spot colour is being used to represent colour values. See COLOUR VALUE, SPOT COLOUR.

colour value: The character string (for spot colour) or values of the point components (for colour model) describing a colour.

RGB: An additive colour model, well matched to colour display monitors, whose values are defined by the normalized weights of Red, Green, and Blue components.

CIE L*u*v*: A colour model defining an absolute colour space based on colour matching experiments whose components are L* (Lightness) and u*, v* (Chromaticness).

CYMK: A subtractive colour model, common in the printing industry, which has cyan, magenta, yellow, and black components.

spot colour: An exactly defined colour with a registered name.

ANSI X3H3

Information Processing Systems --

Computer Graphics --

Metafile for the Storage and Transfer

of Picture Description Information

Part 1

Functional Specification

(Clause 4)

Addendum 3

Draft Document 1.1

June 21, 1988

Page 10

Subclause 4.3: Add the following to the list of elements given in the first paragraph of this clause:

COLOUR REPRESENTATION METHOD

Page 10

Subclause 4.3.2.1: dd the following elements to the list:

CONIC ARC	CONIC ARC TRANSFORMATION MATRIX
PARAMETRIC SPLINE CURVE	RATIONAL B-SPLINE CURVE
PEL ARRAY CLIP RECTANGLE	RATIONAL B-SPLINE CURVE CLOSED
COMPRESSED PEL ARRAY	COMPRESSED PEL ARRAY TILED
PEL ARRAY REFERENCE POINT	

Page 10

Subclause 4.3.2.1: Add the following to the list of elements given in the second paragraph of this clause:

COLOUR REPRESENTATION METHOD

Page 12

Subclause 4.4.2, first line:

Change "direct (RGB) colour"

into "direct colour"

Page 14

Subclause 4.4.6, second paragraph, first line:

Change "RGB"

into "a direct colour"

Page 14

Subclause 4.5.2: add the following to the end of the subclause:

The PEL ARRAY CLIP RECTANGLE is not affected by the setting of the CLIP INDICATOR element. Pel array clipping is assumed to be always 'on'. The default PEL ARRAY CLIP RECTANGLE is listed in clause 6.

Page 15

Subclause 4.6: add the following to the list:

CONIC ARC
PARAMETRIC SPLINE CURVE
RATIONAL B-SPLINE CURVE

RATIONAL B-SPLINE CURVE CLOSED
COMPRESSED PEL ARRAY
COMPRESSED PEL ARRAY TILED

Page 15

Subclause 4.6: add the following to "The line elements are:" list:

CONIC ARC
PARAMETRIC SPLINE CURVE
RATIONAL B-SPLINE CURVE

Page 16

Subclause 4.6: add the following to "The filled-area elements are:" list:

RATIONAL B-SPLINE CURVE CLOSED

Page 16

Subclause 4.6: change "The cell array element is:" to "The cell array elements are:" and add the following to the list:

COMPRESSED PEL ARRAY
COMPRESSED PEL ARRAY TILED

Page 16

Subclause 4.6.1.1: change subclause to read the following:

4.6.1.1 Description. There are two general line elements - POLYLINE and DISJOINT POLYLINE - four line elements relating to circles, ellipses and conic arcs, and two elements that relate to spline curves.

Page 16

Subclause 4.6.1.1: change the end of the subclause:

CONIC ARC	generates a parabolic, hyperbolic or elliptical arc: the parameterization of the arc(s) is described in 5.6.X.
XXX SPLINE CURVE	generates a single spline curve: two separate parameterizations of the spline curve are possible: these are described in 5.6.X and 5.6.X+1

Page 16

Subclause 4.6.1.3: change the last sentence of the subclause to read:

"The ARC and SPLINE primitives...

Page 17

Subclause 4.6.4.1: change the second sentence of the subclause to read:

"In addition there are seven elements that..."

Page 18

Subclause 4.6.4.1: add the following to the end of the subclause:

RATIONAL B-SPLINE CURVE CLOSED	generates a closed B-spline curve: the set of styles is the same as for POLYGON.

Page 18

Subclause 4.6.4.3 2nd paragraph: change the sentence to read:

"The circular, elliptical and B-spline fill primitives..."

Page 18

Subclause 4.6.5: add the following to the list:

COMPRESSED PEL ARRAY XXX	represents a rectangular binary image compressed according to the CCITT T4 or T6 facsimile recommendations: two parameterizations are possible, one corresponding to the normal facsimile-size image, and a tiled format for large images: the elements are described in 5.6.X and 5.6.X+1.

Page 18

Subclause 4.6.5: add the following at the end of the subclause:

4.6.5.1 Clipping. Clipping of the pel array elements is accomplished using the PEL ARRAY CLIP RECTANGLE. Since the pel array clipping 'mode' is always assumed to be 'on', all pel array elements to be displayed can be considered clipped. The default pel array clip rectangle is listed in Clause 6. The PEL ARRAY CLIP RECTANGLE element affects the pel array element that immediately follows it in the metafile, or if no pel array element immediately follows, it affects the last pel array previously defined. In this way, multiple clipped pel arrays can be created by implicitly referencing just one occurence of a pel array element.

4.6.5.2 Positioning. The position of a clipped pel array element is defined by the PEL ARRAY REFERENCE POINT element. The reference point element affects the position of the clipped pel array that immediately follows it in the metafile. If a clipped pel array appears in the metafile without a reference point immediately before it, the array is positioned at the last reference point previously defined. The effect on whatever might already be positioned at a given reference point is implementation-dependent. If a reference point appears with no pel array immediately following it, a duplicate of the last pel array previously defined is postioned and displayed at the reference point.

4.6.5.3 Tiling. For further work, will be based on the Tiled Raster Interchange Format specification (TRIF 2.0). The state of this specification within CALS and 8613 part 7 is too unstable to begin work yet.

Page 19

Subclause 4.6.7: add the following after the subclause:

4.6.8 Conic Arc Element

***NOTE - I have come to the conclusion that the IGES specification upon which this and the transformation element are based are totally unsuited for use in the CGM. This parameterization assumes that the arc is defined in some nebulous definition space that is then transformed to model space. Since the transformation matrix is only 3X3, IGES provides a mechanism to concatenate matrices to eventually arrive at the desired orientation in model space. The CGM has no such mechanism. In addition, the entire specification assumes 3-D space, and changing that to 2-D is not as simple as merely zeroing out the Z term. I would like to withdraw the arcs and splines from this specification, and spend some more time investigating other possible algorithms.

4.6.8.1 Geometric Concepts. The conic arc is defined by the start point, end point and the six parameters A-F. To determine the form of the conic arc, the quantities Q1, Q2 and Q3 are defined as follows:

$$Q1 = \text{determinant of} \begin{vmatrix} A & B/2 & D/2 \\ B/2 & C & E/2 \\ D/2 & E/2 & F \end{vmatrix}$$

$$Q2 = \text{determinant of} \begin{vmatrix} A & B/2 \\ B/2 & C \end{vmatrix}$$

$$Q3 = A + C$$

If $Q2>0$ and $(Q1*Q3)<0$, then the arc is elliptical;
if $Q2<0$ and $Q1<>0$, then the arc is hyperbolic;
if $Q2=0$ and $Q1<>0$, then the arc is parabolic.

In the case where the conic arc is elliptical, to distinguish the arc in question from its compliment, the direction of the arc with respect to VDC space must be from start point to end point in a counterclockwise direction.

In the case where the conic arc is parabolic or hyperbolic, the parameterization defines a unique portion of the parabola or a unique portion of a branch of the hyperbola, thus the direction is irrelevant.

4.6.8.2 Parameterization. A conic arc is defined by the end points and the six parameters. The conic arc itself is defined by the six parameters in the following equation:

$$A(X**2) + BXY + C(Y**2) + DX + EY + F = 0$$

This parameterization assumes that VDC space is 4 quadrant cartesian coordinant space. The CONIC ARC TRANSFORMATION MATRIX element is then used to properly position the arc in the CGM one quadrant VDC picture space.

4.6.9 Spline Curve Elements

4.6.9.1 Parametric Spline Curve. The parametric spline curve is a sequence of parametric polynomial segments. The definition of this class of curves generalized to allow for the representation of many different parametric spline curves using only one element. The following curve types have been assigned:

1: linear
2: quadratic
3: cubic
4: Wilson-Fowler
5: modified Wilson-Fowler
6: B spline

4.6.9.1.1 Parameterization. The degree of continuity parameter indicates the smoothness, or continuity of the curve with respect to arc length. The curve can either be continuous at all break points, continuous and have slope continuity at all break points, or be continuous and have both slope and curvature continuity at all break points.

The number of segments parameter is the number of polynomial segments to be used to define the curve. Each X,Y polynomial segment is evaluated using the eight polynomial coefficients associated with that segment (AX,BX,CX,DX,AY,BY,CY,DY). Each segment is delimited by its respective breakpoint.

4.6.9.1.2 Geometric Concepts. The following cubic polynomial equations will return the coordinates of the points of the i-th segment of the curve. Note that the coefficients D, or C and D will be zero if the polynomials are of degrees 2 or 1, respectively:

$$X(u) = AX(i) + BX(i)(s) + CX(i)(s**2) + DX(i)(s**3)$$
$$Y(u) = AY(i) + BY(i)(s) + CY(i)(s**2) + DY(i)(s**3)$$

where $T(i) <= u <= T(i+1), i=1,...,N$ and $s = u - T(i)$.

The terminate point and derivatives are derived without computing the polynomials by evaluating the Nth polynomials and derivatives at $u = T(N + 1)$. These data, divided by the appropriate factorial (i.e. the second derivative divided by 2!, the third by 3!), are used as the N+1 or terminate point values.

4.6.9.2 Rational B-Spline Curve. The parametric equation governing the definition of the rational B-spline curve is shown in the following expression:

$$\frac{W(0)*P(0)*b0(t)}{W(0)*b0(t)} + \frac{W(1)*P(1)*b1(t)}{W(1)*b1(t)} + \ldots + \frac{W(K)*P(K)*bK(t)}{W(K)*bK(t)}$$

where W(i) are the weights, P(i) are the control points and bi(t) are the basis functions. The basis functions are all non-negative piecewise polynomials determined by the degree and the knot sequence. The knot sequence is a non-decreasing list of real numbers T(-M),...T(0),...T(K+1). Each basis function is supported for the knot sequence interval [T(i-M),T(i+1)], where M is the dgree of the basis function. Between any two adjacent knot values, the corresponding basis function can be expressed as a single polynomial.

The curve itself is parameterized where:

```
start_param <= t <= end_param,
T(0) <= start_param < end_param <= T(N)
```

Thus for any parameter value t between T(0) and T(K+1), the sum of the basis functions satisfies the following identity:

```
b0(t) + b1(t) + .... + bK(t) = 1.
```

If all of the weights in the weight list are not equal, then the equation type is rational. Otherwise, if all of the weights are equal, then all of the weights cancel, the denominators sum to one and the equation type is polynomial.

4.X.X Compound Line

The compound line elements consist of the two elements, BEGIN COMPOUND LINE and END COMPOUND LINE. These elements permit the definition of a line that consists of a number of distinct elements, such as straight lines and arcs, which is treated as if it were a single line element. Thus, for example, line style would apply without change or interruption past a straight line segment onto a following arc segment. Likewise, the ends of the various component elements of the compound line are not treated as line end.caps but rather as line joints.

4.X.X Compound Text Path

The compound text path elements consist of the two elements, BEGIN COMPOUND TEXT PATH and END COMPOUND TEXT PATH. It is functionally identical to Compound Line, except that it is used as a base line for text placement, rather than drawn by an interpreter.

The Compound Text Path permits arbitrary, complex placement of text. Each font symbol is placed with its reference point and alignment according to a tangent to the Compound Text Path. This implicit tangent is the logical base line for each character cell. If a symbol's reference point aligns with the junction of two line elements of the Compound Text Path, the logical base line is the line perpendicular to the perpendicular bisector of the tangents of both elements, passing through the reference point. Positioning of

subseuent symbols is based upon the distance between symbols assuming a straight base line, but wrapped along the generalized curve of the Compound Text Path. If there is more text than path, the path for the excess text is the straight line described by the tangent to the last element of the Compound Text Path.

4.X.X Picture Composition

The picture composition elements consist of BEGIN CLIP REGION, END CLIP REGION, BEGIN SHIELD REGION, END SHIELD REGION, CLIP INDICATOR, and SHIELD INDICATOR.

The concepts of clip and shield regions are complementary. The clipping process discards everything that is visually outside the clip region whereas the shielding process discards everything that is inside the shield region. Whether clipping and shielding are in effect is determined by the respective settings of the CLIP INDICATOR and SHIELD INDICATOR (each is either ON or OFF).

Due to being able to define what amounts to closed figures for these regions, the clip region and shield regions can each provide a clipping and a shielding capability at the same time. For example, a "polygon with a hole" has an outer boundary and an inner boundary. The "filled area" of such a clipping region would be preserved with the area outside the "filled area" (including the contents of the hole") being removed from the picture. The shielding region has a complementary interpretation: the "filled area" itself is removed from the picture.

Note that the shielding effect of a clip region "hole" is independent of the SHIELD INDICATOR and likewise, the clipping effect of a shield region "hole" is independent of the CLIP INDICATOR.

Page 37

Subclause 4.7.7: replace the current text body with the following:

The CGM must be able to represent colours in a manner suitable to a broad spectrum of graphical devices and systems. Towards this goal, the CGM uses three colour models (RGB, CIE L*u*v*, CYMK) and spot colour. The Metafile Descriptor element, COLOUR REPRESENTATION METHOD, allows metafile generators to specify which one is being employed.

The RGB additive colour model uses a 3-tuple of values providing the normalized weight of the red (R), green (G), and blue (B) components of the desired colour. This model is used in colour monitors, film writers, and input scanners. The source and destination graphical devices must be "correctly" calibrated in order for a point in this model to be perceived as the same colour on both devices.

The CIE L*u*v* uniform colour model uses a 3-tuple of values providing the normalized luminance (L*), red-green chromaticness (u*), and blue-yellow chromaticness (v*) components of the desired colour. The advantages of the model over the others are (1) it separates chromaticness from luminance (allowing for easy monochrome display), (2) it is a uniform space for small colour differences (the Euclidean distance between two points in this model is more or less proportional to their perceived difference), and (3) it is an absolute colour model based colour matching experiments (thus being

device independent and not requiring colour correction).

The CYMK subtractive colour model uses a 4-tuple of values providing the normalized weights of the cyan (C), yellow (Y), magenta (M), and black (K) components of the desired colour. This model is used by printers and Graphics Arts drum scanners. In theory, cyan, magenta, and yellow are meant to correspond to the red, green, and blue of the RGB model. In practice, actual inks used for colour printing only approximate this criterion. Black ink is added to attain a greater dynamic range than is possible with three colours alone. In particular, it allows the attainment of a much richer black than would be possible using only the first three inks.

Spot colour uses a character string representing a registered or private colour name (similar in format to named fonts). Use of the former is recommended for metafile transportability, because registration ensures unique naming of colours. Spot colours are registered in the ISO International Register of Graphical Items, which is maintained by the Registration Authority. When a spot colour has been approved by the ISO Working Group on Computer Graphics, the colour name will be assigned by the Registration Authority. Each registered colour will be specified in terms of its CIE L*u*v* coordinates.

The CGM provides two mechanisms for colour selection: 'direct' and 'indexed'. In 'direct' colour selection, the colour is specified by providing values for its normalized components (colour model) or by providing the character string defining its name (spot colour). (The term 'direct colour value' will refer to any direct colour specifier, and the term 'direct colour model value' will refer only to a direct colour specifier of a colour model (as opposed to spot colour)). In 'indexed' colour selection, the colour is specified by an index into a table of direct colour values. Selection of one of these mechanisms may be done by an element in each Picture Descriptor.

For 'indexed' colour selection, the COLOUR TABLE attribute element is provided for changing the contents of the colour table. This element may appear throughout the picture body. However, the effect of changes in the colour table on any existing graphical primitive elements that use the affected indices is not addressed in this Standard.

Direct colour model values are either a 3-tuple or 4-tuple of values providing the normalized weight of the desired colour components. In the abstract, each component is normalized to the continuous range of real numbers [0,1]. In a metafile, direct colour model value components are integers, and the Metafile Descriptor element, COLOUR VALUE EXTENT, allows metafile generators to specify the minimum and maximum metafile direct colour model values for normalization.

ANSI X3H3

Information Processing Systems --

Computer Graphics --

Metafile for the Storage and Transfer of Picture Description Information

Part 1

Functional Specification

(Clause 5)

Addendum 3

Draft Document 1.1

June 21, 1988

Page 41

Sub-clause 5.1: Table of data type abbreviations:

Replace

CD	Colour Direct	Three-tuple of non-negative real values for red, green, blue colour intensities.

with

MD	Model Direct	Three-tuple or four-tuple of non-negative real values for colour definition within one of the three supported colour models.
CD	Colour Direct	String or Model Direct (as determined by COLOUR REPRESENTATION METHOD).

Pages 47-48

Sub-clause 5.3.7: Replace the description as follows:

The precision for operands of data type model direct (MD) is specified for subsequent data of type MD. The precision is defined as the field width measured in units applicable to the specific encoding.

Although the form of the parameter is encoding dependent, the parameter is a single specification that applies to each or all of the three or four components of parameters of type CM.. The precisions of the individual components are not independently and differently specifiable by this element.

Pages 48-49

Sub-clause 5.3.10: Replace the description as follows:

The parameters represent an extent which bounds the direct colour model values that will be encountered in the metafile. It need not represent the exact extent of colour model values contained in the metafile.

Page 54

Subclause 5.3: Add the following Metafile Descripter Elements:

5.3.X COLOUR REPRESENTATION METHOD

Parameters:

colour representation method (one of: RGB, CIE L*u*v*, CYMK, spot colour) (E)

Description:

Four methods of colour representation are supported: by one of three colour models (RGB, CIE L*u*v*, CYMK) or by spot colour (names).

Only one colour representation method may be used within a metafile.

The method may be defaulted or explicitly set with the COLOUR REPRESENTATION METHOD element. All occurrences of colour-setting elements (AUXILIARY COLOUR, LINE COLOUR, MARKER COLOUR, FILL COLOUR, EDGE COLOUR, TEXT COLOUR) as well as the colour lists of CELL ARRAY and PATTERN TABLE shall be in the current method. If used, COLOUR REPRESENTATION METHOD shall be in the Metafile Descriptor, after BEGIN METAFILE and before BEGIN METAFILE BODY.

References:

4.7.7

5.3.X FONTMETRIC DEFINITION

Parameters:

font index (IX)
character index (C)
left bearing (format to be determined)
right bearing (format to be determined)
character height (format to be determined)
offset from baseline (format to be determined)

Description:
The fontmetric information for each character used in each font specified is defined by this element. If this element is used, then the fontmetric data for each character used in the metafile must be specified. Characters not used by the metafile may also be specified, but are not required.

References:

5.3.X CHARACTER KERNING MODE

Parameters:

character kerning mode (one of: none, pair, sectored) (E)

Description:
Defines the kerning style, if any, for the metafile.

References:

5.3.X CHARACTER KERNING TABLE

Parameters:

To be determined.

Description:
The data defined by this element will be dependant upon which, if any, kerning styles are supported. In general, however, the information will be that which is required to kern characters.

References:

5.3.X FCS TYPE

Parameters:

Font coordinate type (one of: integer, real, VDC) (E)

Description:
The single parameter is an enumerated value that declares the data type, integer or real, of the font coordinate space. Font coordinate space may be different than VDC space because higher precision may be necessary to accurately define the fontmetric data. However, if the font coordinate type is VDC, then the font coordinate space will map to VDC space. In this case, no additional specification for font coordinates (FCS INTEGER PRECISION, FCS REAL PRECISION, or FCS EXTENT) need be specified.

References:

5.3.X FCS INTEGER PRECISION

Parameters:

Encoding dependant.

Description:
The indicated integer precision for fontmetric data. The precision is defined as the field width measured in units applicable to the specific encoding.

References:

5.3.X FCS REAL PRECISION

Parameters:

Encoding dependant.

Description:
The indicated real precision for fontmetric data. The precision is defined as the field width measured in units applicable to the specific encoding.

References:

5.3.X CAP HEIGHT

Parameters:

To be determined.

Description:
To be determined.

References:

5.3.X FCS EXTENTS

Parameters:

first corner (P)
second corner (P)

Description:
The two corners define a rectangular extent in font coordinate space that demarcates a two-dimensional Cartesian coordinate system. Each character definition within a font must be defined completely within this coordinate system.

*** NOTE *** The need for this element is specified in ISO 9541-5, clause 4. The intent of this element is to address the needs outlined by ISO. Additional works must be done.

References:

Page 55

Sub-clause 5.4.2, first paragraph of description:

Change "red, green, and blue" into "direct"

Page 57

Sub-clause 5.4.7, first line of second paragraph of description:

Change "RGB" into "a direct colour value"

Page 60

Subclause 5.5: Add the following Control Elements:

5.5.X BEGIN COMPOUND LINE

Parameters:

none

Description:
BEGIN COMPOUND LINE delimits the beginning of a definition of a entity that will have consistent line attributes and will be treated as a single "compound primitive". The elements that make up the compound line can be any combination of non-closed line elements such as POLYLINE, CIRCULAR ARC 3 POINT, CIRCULAR ARC CENTRE, ELLIPTICAL ARC, or <new curve elements>.

References:

5.5.X END COMPOUND LINE

Parameters:

none

Description:
END COMPOUND LINE delimits the end of a compound line definition.

References:

5.5.X BEGIN COMPOUND TEXT PATH

Parameters:

none

Description:
BEGIN COMPOUND TEXT PATH delimits the beginning of a definition of a entity that will provide the path in which a text string will be drawn. The elements that make up the compound text path can be any combination of non-closed line elements such as POLYLINE, DISJOINT POLYLINE, CIRCULAR ARC 3 POINT, CIRCULAR ARC CENTRE, ELLIPTICAL ARC, or <new curve elements>.

Once defined, the compound text path takes the place of the text path as defined by the TEXT PATH element and the CHARACTER ORIENTATION elements. The skew of the characters is still relative to that specified in the CHARACTER ORIENTATION element, but the placement of subsequent characters is along the compound text path instead of in a line along the character up vector or character base vector.

References:

5.5.X END COMPOUND TEXT PATH

Parameters:

none

Description:
END COMPOUND TEXT PATH delimits the end of a compound text path definition.

References:

5.5.X BEGIN CLIP REGION

Parameters:

none

Description:
BEGIN CLIP REGION delimits the beginning of a definition of a entity that will provide the clipping region. When CLIP INDICATOR is 'on' only the portions of graphics elements inside or on the boundary of the clipping region are drawn. The elements that make up the clipping region can be any combination of closed or non-closed elements such as POLYLINE, DISJOINT POLYLINE, POLYGON, POLYGON SET, CIRCULAR ARC 3 POINT, CIRCULAR ARC 3 POINT CLOSE, CIRCULAR ARC CENTRE, CIRCULAR ARC

CENTRE CLOSE, ELLIPTICAL ARC CLOSE, or <new curve elements>. The entity thus defined is essentially a closed figure whose boundary is used as the clipping boundary.

Once defined, the clipping region takes the place of the clipping region defined in CLIP RECTANGLE.

References:

5.5.X END CLIP REGION

Parameters:

none

Description:

END CLIP REGION delimits the end of a clipping region definition.

References:

5.5.X BEGIN SHIELD REGION

Parameters:

none

Description:

BEGIN SHIELD REGION delimits the beginning of a definition of a entity that will provide the shielding region. When SHIELD INDICATOR is 'on' only the portions of graphics elements outside of the shielding region are drawn. The elements that make up the shielding region can be any combination of closed or non-closed elements such as POLYLINE, DISJOINT POLYLINE, POLYGON, POLYGON SET, CIRCULAR ARC 3 POINT, CIRCULAR ARC 3 POINT CLOSE, CIRCULAR ARC CENTRE, CIRCULAR ARC CENTRE CLOSE, ELLIPTICAL ARC CLOSE, or <new curve elements>. The entity thus defined is essentially a closed figure whose boundary is used as the shielding boundary.

References:

5.5.X END SHIELD REGION

Parameters:

none

Description:

END SHIELD REGION delimits the end of a shielding region definition.

References:

5.5.X SHIELDING INDICATOR

Parameters:

shield indicator (one of: off, on) (E)

Description:

When SHIELD INDICATOR is 'off', shielding of graphical primitive elements is not required.

When SHIELD INDICATOR is 'on', only those portions of graphical primitive elements outside of the shielding region are drawn.

References:

Page 68

Sub-clause 5.6.9: Add the following at the end of the third paragraph of the description:

Note that COLOUR PRECISION only applies to direct colour model values.

Page 77

Subclause 5.6: Add the following Graphical Primitive Elements:

5.6.X CONIC ARC ***PRELIMINARY***

Parameters:

start point (P)
end point (p)
A,B,C,D,E,F (6R)

Description:

A conic arc is drawn which is defined as follows:

A conic arc is defined by the end points and the six parameters. The conic arc itself is defined by the six parameters in the following equation:

$$A(X^{**}2) + BXY + C(Y^{**}2) + DX + EY + F = 0$$

In order for the conic arc to be processed correctly by the receiving system given the above representation, the conic arc entity must be positioned such that each of its axes is parallel to either the XT axis or YT axis. The arc is then positioned correctly in VDC space by using the value of the CONIC ARC TRANSFORMATION MATRIX element.

To determine the form of the conic arc, the quantities Q1, Q2 and Q3 are defined as follows:

$$Q1 = \text{determinant of} \begin{vmatrix} A & B/2 & D/2 \\ B/2 & C & E/2 \\ D/2 & E/2 & F \end{vmatrix}$$

$$Q2 = \text{determinant of } \begin{vmatrix} A & B/2 \\ B/2 & C \end{vmatrix}$$

$$Q3 = A + C$$

If Q2>0 and (Q1*Q3)<0, then the arc is an ellipse.
If Q2<0 and Q1<>0, then the arc is a hyperbola.
If Q2=0 and Q1<>0, then the arc is a parabola.

In the case where the conic arc is elliptical, to distinguish the arc in question from its compliment, the direction of the arc with respect to the definition space must be from start point to end point in a counterclockwise direction.

In the case where the conic arc is parabolic or hyperbolic, the parameterization defines a unique portion of the parabola or a unique portion of a branch of the hyperbola, thus the direction is irrelevant.

The direction of the conic arc with respect to VDC space is determined by the original direction of the arc in definition space, in conjunction with the action of the CONIC ARC TRANSFORMATION MATRIX element.

References:

4.

5.6.X CONIC ARC TRANSFORMATION MATRIX ***PRELIMINARY***

Parameters:

matrix elements

if the VDC type is 'integer',
R11,R12,R21,R22 (4I)

if the VDC type is 'real',
R11,R12,R21,R22 (4R)

coordinate offset (P)

Description:

This element is intended to work in conjunction with the CONIC ARC element to transform the conic arc from definition space to VDC space. The Transformation Matrix entity transforms the definition space point coordinates by means of a matrix multiplication and then the addition of an offset.

The notation for this transformation is as follows:

$$\begin{vmatrix} R11 & R12 \\ R21 & R22 \end{vmatrix} \begin{vmatrix} Xin \\ Yin \end{vmatrix} + \begin{vmatrix} T1 \\ T2 \end{vmatrix} = \begin{vmatrix} Xout \\ Yout \end{vmatrix}$$

where |Rij| is the transformation matrix. (Xin,Yin) is the coordinate to be transformed, (T1,T2) is the offset, and (Xout,Yout) is the coordinate resulting from the transformation. Both the input and output coordinate systems are assumed to be orthogonal, cartesian and right-handed.

References:
4.

5.6.X PARAMETRIC SPLINE CURVE

Parameters:

curve_type (IX)
H-degree of continuity (I)
N-number of segments (I)
T-break point list for polynomial ((N+1)R)
X coordinate polynomial list (N sets of four)

AX,BX,CX,DX (N*4R)

Y coordinate polynomial list (N sets of four)

AY,BY,CY,DY (N*4R)

Description:
The parametric curve to be drawn is defined as follows:

The parametric spline curve is a sequence of parametric polynomial segments. The parameterization shown above is generalized to allow for the representation of many different parametric spline curves using this one element. The curve_type parameter indicates the type of parametric curve as it was represented in the sending system before being converted to this generic form. The following curve types have been assigned:

1: linear
2: quadratic
3: cubic
4: Wilson-Fowler
5: modified Wilson-Fowler
6: B spline

Values above 6 are reserved for registration and future standardization, and negative values are available for implementation-dependent use.

The degree of continuity parameter, H, indicates the smoothness, or continuity of the curve with respect to arc length. If H=0, the curve is continuous at all break points. If H=1, the curve is continuous and has slope continuity at all break points. If H=2, the curve is continuous and has both slope and curvature continuity at all break points.

The number of segments parameter, N, is the number of polynomial segments to be used to define the curve.

The parametric spline curve is displayed with the current line attributes.

*** Additional info. on degeneracies required ***

References:
4.

5.6.X RATIONAL B-SPLINE CURVE

Parameters:

K-upper index of sum (I)
M-degree of the basis function (I)
equation_type flag (one of: rational, polynomial) (E)
periodic flag (one of: non-periodic, periodic) (E)
T-knot sequence list ((K+M+2)R)
W-weight list ((K+1)R)
control point list ((K+1)P)
start_param,end_param (2R)

Description:
A rational spline curve is drawn which is defined as follows:

*** Additional info. required. ***

Valid values of the upper index and degree paramenters are non-negative integers.

Valid values of the control points are such that no two adjacent points are coincident.

If all of the weights in the weight list W are not equal, then the equation_type flag is set to Rational, otherwise the equation_type flag is set to Polynomial.

References:
4.

5.6.x COMPRESSED PEL ARRAY

Parameters:

T-encoding type (one of:T4,T6) (E)
P-pel path (one of:0,90,180,270) (E)
L-line progression (one of:90,270) (E)
S-pel spacing (R)
spacing_ratio (R)
N-number of pels per line (I)
NL-number of lines (I)
pel array (BS)

Description:
A compressed pel array image is defined as follows:

The encoding type parameter, T, specifies the CCITT compression format used to encode the image. If T is specified as "T4", the image is encoded according the one or two dimensional scheme defined in CCITT Recommendation T.4 (Group 3 facsimile). If T is "T6", the image is encoded according to the two dimensional scheme defined in CCITT Recommendation T.6 (Group 4 facsimile).

The position of the upper left-hand corner of the clipped pel array is defined by the reference point (See PEL ARRAY CLIP RECTANGLE).

The pel path parameter, P, is the direction of progression of successive

pels along a line relative to the VDC X axis. This parameter, in conjunction with the pel spacing, S, and the number of pels per line, N, implicitly define the line position, length and granularity for each line in the decoded pel array. The spacing_ratio is defined as the ratio M/S, where "M" is the distance in Basic Measurement Units (BMUs - 1200 BMUs per inch) between two successive lines of pels, and "S" is the distance in BMUs between two adjacent pels in a line.

The line progression parameter, L, is the direction of progression of successive of pel lines and is expressed as a direction relative to P. L, in conjunction with the spacing_ratio and the number of lines, NL, implicitly defines the size of the decoded image in the direction of L. The line spacing (LS) of the lines of pels can be determined as follows:

LS = spacing_ratio * S

The pel array itself is stored in the formats defined by T, encoded as a bit_stream.

References:
4.

5.6.x PEL ARRAY TILED ***PRELIMINARY***

Parameters:

T-encoding type (one of:bitmap,T4,T6) (E)
P-pel path (one of:0,90,180,270) (E)
L-line progression (one of:90,270) (E)
S-pel spacing (R)
spacing_ratio (R)
NT-number of tiles (I)
N-number of pels per line (I)
NL-number of lines (I)
TD-tile dimensions (2I)

pel array (BS)

Description:
A tiled pel array image is defined as follows:

*** Additional specification required. ***

References:

5.6.x PEL ARRAY CLIP RECTANGLE

Parameters:

X1,Y1,X2,Y2 (4I)
first corner (P)
second corner (P)

Description:
The element defines the rectangular area of pels in the decoded pel array that is to appear in the metafile picture. This element should

immediately precede the COMPRESSED PEL ARRAY element to which it applies, as this element will affect all subsequent COMPRESSED PEL ARRAY elements until another PEL ARRAY CLIP RECTANGLE is explicitly defined.

The four integers form two coordinate pairs, (X1,Y1) and (X2,Y2) corresponding to the first and last pels to appear, respectively, where X1<=X2 and Y1<=Y2. For example, (6,2) would specify the seventh pel in line 3, given that (0,0) specifies the first pel on the first line.

The two corner points define the clipped pel array to be positioned in VDC space. The first pel defined by (X1,Y1) above is to be positioned at the reference point of the COMPRESSED PEL ARRAY element. Only those portions of the decoded pel array from the COMPRESSED PEL ARRAY element inside or on the boundary of the pel array clip rectangle are to be rendered.

References:
4.

5.6.X PEL ARRAY REFERENCE POINT

Parameters:

reference_point (P)

Description:
The pel array reference point defines the upper left-hand corner of the pel array element to be displayed. If the pel path and line progression are thought of as vectors, the upper left-hand corner is defined as point of origin for the two vectors.

References:
4.

Page 97

Subclause 5.7: Add the following attribute elements:

5.7.X LINE TYPE DEFINITION

Parameters:

linetype (IX)
dash unit selector (one of: VDC, mm, native device units, abstract) (E)
dash repeat length (R)
adaptive flag (one of: no, yes) (E)
list of dash elements (nI)

Description:
This element defines a linetype and associates it with an index for future reference. Parameter 'linetype' is the index of linetype being defined. The parameter 'list of dash elements' is the definition to be associated with the index. The first element is a dash, second a space, etc. -- the defined linetype is solid for I1 units, gap for I2 units, solid for I3 units, etc. N must be positive, and each dash element (I) non-negative. N=1 means a solid line: I=0 interpreted as a dot.

The units of the 'dash repeat length' are specified by the 'dash unit selector' parameter. The value of 'abstract' indicates that the implementation may normalize and map the sum of the dash pattern elements at its discretion. The 'dash repeat length' defines the length of one complete cycle of the dash pattern, measured in the units of 'dash unit selector'.

An "adaptive" linetype is one where every vertex falls on an inked portion of the line. This is accomplished in plotters by temporarily modifying the duty cycle for each line segment (ceiling function) such that there is always an integral number of repeats (and all predefined linetypes have their gaps_array defined such that they begin and end with inked or "pen down" portions).

References:

5.7.X HATCH STYLE DEFINITION

Parameters:

hatch index (IX)
style indicator (one of: parallel, crosshatch) (E)
hatch space units selector (one of: VDC, mm, device units, abstract) (E)
angle (2R)
duty cycle length (R)
list of hatch elements (nI) - I>=0, n>=2

Description:
This element defines a hatch style and associates it with an index for future reference.

The 'hatch index' parameter defines the index of hatch style being defined. The 'list of hatch elements' is an array that defines alternating line width and gap width -- i.e., the width of a hatch line followed by the width of the space to the next hatch line. The center of the first hatch line is matched up with PATTERN REFERENCE POINT, if implemented. 0 interpreted as thinnest line width available.

The 'hatch space units selector' specifies the units of 'duty cycle length'. It also controls the manner of transformation of the hatching: If VDC, then the hatching transforms with segment transform and anisotropic transforms (as if hatching had done POLYLINES); otherwise, the hatching is like "wallpaper" that shows through the polygon-shaped hole -- everything is defined in device units and hatching performed in device space. The value of 'abstract' indicates that the implementation may normalize and map the sum of the list of hatch elements at its discretion. The 'duty cycle length' is measured perpendicular to the hatch line. The sum of hatch elements in the hatch element list is normalized to this distance before presentation of the hatch on the view surface.

The 'angle' parameter is measured in the units specified by the 'hatch space units selector'. It consists of two components, dx and dy, defining a vector.

5.7.X LINE CAP

Parameters:

line cap indicator (one of: butt, round, projecting square) (E)

Description:
The line cap style is defined for subsequent line elements. The line cap style determines the appearance of open endpoints (as opposed to interior vertices) of line elements. The defined styles are:

butt cap: the line is squared off at the endpoint, there is no projection beyond the endpoint.

round cap: a semicircular arc with diameter equal to the line width is drawn around the endpoint and filled in. The drawn line thus projects beyond the endpoint.

projecting square cap: the line is squared off at a distance equal to half the line width beyond the endpoint.

References:

5.7.X LINE JOIN

Parameters:

line join indicator (one of: miter, round, bevel) (E)

Description:
The line join style is defined for subsequent line elements. The line join style defines the appearance of interior vertices of polyline elements and of compound line elements. The defined styles are:

miter join: the outer edges of the two adjoining line segments are extended until they meet at a point.

round join: a circular arc with diameter equal to the line width is drawn around the vertex between the adjoining segments and is filled in, producing a rounded corner.

bevel join: the adjoining line segments are terminated with a butt cap, and the resulting triangular notch is filled in.

References:

5.7.X EDGE CAP

Parameters:

edge cap indicator (one of: butt, round, projecting square) (E)

Description:
The edge cap style is defined for subsequent edge elements. The edge cap style determines the appearance of open endpoints of filled area edges (such as may result from a mixture of visible and invisible edge segments). The defined styles are:

butt cap: the edge is squared off at the vertex, there is no projection beyond the endpoint.

round cap: a semicircular arc with diameter equal to the edge width is drawn around the endpoint and filled in. The drawn edge thus projects beyond the endpoint.

projecting square cap: the edge is squared off at a distance equal to half the edge width beyond the endpoint.

References:

5.7.X EDGE JOIN

Parameters:

edge join indicator (one of: miter, round, bevel) (E)

Description:

The edge join style is defined for subsequent filled elements. The edge join style defines the appearance of interior vertices of filled area elements. The defined styles are:

miter join: the outer edges of the two adjoining edge segments are extended until they meet at a point.

round join: a circular arc with diameter equal to the edge width is drawn around the vertex between the adjoining segments and is filled in, producing a rounded corner.

bevel join: the adjoining edge segments are terminated with a butt cap, and the resulting triangular notch is filled in.

References:

5.7.X MITER LIMIT

Parameters:

miter limit (R)

Description:

Mitered corners can extend very far beyond the line vertex if the angle between the adjoining line segments is small. Miter length is defined to be the distance from the point at which the inner edges of the adjoining line segments meet to the point at which the outer edges meet. If miter length exceeds the 'miter limit' parameter, then the joining line segments are rendered with a bevel join instead of a miter join.

Miter limit is measured as a scale factor applied to the current line width. Miter limit applies to line elements and edges of filled areas.

References:

5.7.X EXTERNAL SYMBOL

Parameters:

To be determined.

Description:
Reference to external defined symbol libraries is provided. The mechanism of this element is yet to be defined.

Page 95

Sub-clause 5.7.32: Add the following at the end of the third paragraph of the description:

Note that COLOUR PRECISION only applies to direct colour model values.

ANSI X3H3

Information Processing Systems --

Computer Graphics --

Metafile for the Storage and Transfer
of Picture Description Information

Part 2

Character Encoding

Addendum 3

Draft Document 1.1

June 21, 1988

Page 12

Clause 5, Table 1: Add the following opcodes to Table 1:

TABLE 1. Opcodes for Metafile Elements.

Opcode	7-Bit Coding		8-Bit Coding	
FONTMETRIC DEFINITION opcode	3/1	3/0	03/1	03/0
CHARACTER KERNING MODE opcode	3/1	3/1	03/1	03/1
CHARACTER KERNING TABLE opcode	3/1	3/2	03/1	03/2
FCS TYPE opcode	3/1	3/3	03/1	03/3
FCS INTEGER PRECISION opcode	3/1	3/4	03/1	03/4
FCS REAL PRECISION opcode	3/1	3/5	03/1	03/5
FCS EXTENT opcode	3/1	3/6	03/1	03/6
CAP HEIGHT opcode	TBD		TBD	
BEGIN COMPOUND LINE opcode	3/3	3/6	03/3	03/6
END COMPOUND LINE opcode	3/3	3/7	03/3	03/7
BEGIN COMPOUND TEXT PATH opcode	3/3	3/8	03/3	03/8
END COMPOUND TEXT PATH opcode	3/3	3/9	03/3	03/9
BEGIN CLIP REGION opcode	3/3	3/10	03/3	03/10
END CLIP REGION opcode	3/3	3/11	03/3	03/11
BEGIN SHIELD REGION opcode	3/3	3/12	03/3	03/12
END SHIELD REGION opcode	3/3	3/13	03/3	03/13
SHIELDING INDICATOR opcode	3/3	3/14	03/3	03/14
CONIC ARC opcode	3/4	3/0	03/4	03/0
CONIC ARC TRANSFORMATION opcode	3/4	3/1	03/4	03/1
PARAMETRIC SPLINE CURVE opcode	3/4	3/2	03/4	03/2
RATIONAL B-SPLINE CURVE opcode	3/4	3/3	03/4	03/3
COMPRESSED PEL ARRAY opcode	3/4	3/4	03/4	03/4
PEL ARRAY CLIP RECTANGLE opcode	3/4	3/5	03/4	03/5
LINE TYPE DEFINITION opcode	3/6	3/4	03/6	03/4
HATCH STYLE DEFINITION opcode	3/6	3/5	03/6	03/5
LINE CAP opcode	3/6	3/6	03/6	03/6
LINE JOIN opcode	3/6	3/7	03/6	03/7
EDGE CAP opcode	3/6	3/8	03/6	03/8
EDGE JOIN opcode	3/6	3/9	03/6	03/9
MITER LIMIT opcode	3/6	3/10	03/6	03/10
EXTERNAL SYMBOL opcode	3/6	3/11	03/6	03/11

Page 33

Subclause 8.2: Add the following element representations:

8.2.x FONTMETRIC DEFINITION

```
<FONTMETRIC-DEFINITION-opcode: 3/1 3/0>
<integer: font-index>
<xxxxx: character-index>
<xxxxx: left-bearing>
<xxxxx: right-bearing>
<xxxxx: character-height>
<xxxxx: baseline-offset>
```

8.2.x CHARACTER KERNING MODE

```
<CHARACTER-KERNING-MODE-opcode: 3/1 3/1>
<enumerated: character-kerning-mode>

<enumerated: character-kerning-mode>  =  <integer: 0> (NONE)
                                      |  <integer: 1> (PAIR)
                                      |  <integer: 2> (SECTORED)
```

8.2.x CHARACTER KERNING TABLE

```
<CHARACTER-KERNING-TABLE-opcode: 3/1 3/2>
<xxxxx: to be determined>
```

8.2.x FCS TYPE

```
<FCS-TYPE-opcode: 3/1 3/3>
<enumerated: FCS-type>

<enumerated: FCS-type>  =  <integer: 0> (INTEGER)
                        |  <integer: 1> (REAL)
                        |  <integer: 2> (VDC)
```

8.2.x FCS INTEGER PRECISION

```
<FCS-INTEGER-PRECISION-opcode: 3/1 3/4>
<integer: largest-integer-code + 1>
```

The largest-integer-code tells how many bits occur in the largest posible magnitude for an integer. For example, if integers in the metafile can range from -32767 to +32767, the largest-integer-code is 15. One aditional bit is required for the sign, and so is added to obtain the proper precision. Thus in this example the parameter would be 16.

8.2.x FCS REAL PRECISION

```
<FCS-REAL-PRECISION-opcode: 3/1 3/5>
<integer: largest-real-code + 1>
<integer: smallest-real-code>
<integer: default-exponent-for-reals>
<enumerated: exponents-allowed>

<enumerated: exponents-allowed> = <integer: 0> (allowed)
                                | <integer: 1> (forbidden)
```

See 8.2.5 of ANSI X3.122-1986 for a description.

8.2.x FCS EXTENT

```
<FCS-EXTENT-opcode: 3/1 3/6>
<fcspoint: first-corner>
<fcspoint: second-corner>

<fcspoint> ::= <integer><integer>
             | <real><real>
```

where a point list is encoded such that the first point is absolute within font coordinate space and each of the following points is relative to its previous one.

8.2.X CAP HEIGTH

<TBD>

Page 40

Subclause 8.4: Add the following element representations:

8.4.X BEGIN COMPOUND LINE

<BEGIN-COMPOUND-LINE-opcode: 3/3 3/6>

8.4.X END COMPOUND LINE

<END-COMPOUND-LINE-opcode: 3/3 3/7>

8.4.X BEGIN COMPOUND TEXT PATH

<BEGIN-COMPOUND-TEXT-PATH-opcode: 3/3 3/8>

8.4.X END COMPOUND TEXT PATH

<END-COMPOUND-TEXT-PATH-opcode: 3/3 3/9>

8.4.X BEGIN CLIP REGION

<BEGIN-CLIP-REGION-opcode: 3/3 3/10>

8.4.X END CLIP REGION

<END-CLIP-REGION-opcode: 3/3 3/11>

8.4.X BEGIN SHIELD REGION

<BEGIN-SHIELD-REGION-opcode: 3/3 3/12>

8.4.X END SHILD REGION

<END-SHIELD-REGION-opcode: 3/3 3/13>

8.4.X SHIELDING INDICATOR

```
<SHIELDING-INDICATOR-opcode: 3/3 3/14>
<enumerated: shielding-indicator>

<enumerated: shielding-indicator> = <integer: 0> (off)
                                  | <integer: 1> (on)
```

Subclause 8.5: Add the following element representations:

8.5.X CONIC ARC

```
<CONIC-ARC-opcode: 3/4 3/0>
<point: start-point>
<point: end-point>
<real: first-param>
<real: second-param>
<real: third-param>
<real: fourth-param>
<real: fifth-param>
<real: sixth-param>
```

8.5.X CONIC ARC TRANSFORMATION

```
<CONIC-ARC-TRANSFORMATION-opcode: 3/4 3/1>
<VDC: R11>
<VDC: R12>
<VDC: R21>
<VDC: R22>
```

8.5.X PARAMETRIC SPLINE CURVE

```
<PARAMETRIC-SPLINE-CURVE-opcode: 3/4 3/2>
<integer: curve-type>
<integer: M-degree-of-continuity>
<integer: N-number-of-segments>
<??????????>
<??????????>
<??????????>
```

8.5.X RATIONAL B-SPLINE CURVE

```
<RATIONAL-B-SPLINE-CURVE-opcode: 3/4 3/3>
<integer: upper-index-of-sums>
<integer: M-degree-of-the-basis-function>
<enumerated: curve-open-flag>
<enumerated: equation-type-flag>
<enumerated: periodic-flag>
<???????????>
<???????????>
<???????????>
<real: start-param>
<real: end-param>

<enumerated: curve-open-flag> = <integer: 0> (OPEN)
                              | <integer: 1> (CLOSED)

<enumerated: equation-type-flag> = <integer: 0> (RATIONAL)
                                 | <integer: 1> (POLYNOMIAL)

<enumerated: periodic-flag> = <integer: 0> (NON-PERIODIC)
                            | <integer: 1> (PERIODIC)
```

8.5.X COMPRESSED PEL ARRAY

```
<COMPRESSED-PEL-ARRAY-opcode: 3/4 3/4>
<enumerated: T-encoding>
<enumerated: P-pel-path>
<enumerated: L-line-progression>
<real: S-pel-spacing>
<real: spacing-ratio>
<integer: N-number-of-pels-per-line>
<integer: NL-number-of-lines>
<xxxxxxx: pel-array>

<enumerated: T-encoding>  = <integer: 0> (T4)
                          | <integer: 1> (T6)

<enumerated: P-pel-path> = <integer: 0> (0-DEGREES)
                         | <integer: 1> (90-DEGREES)
                         | <integer: 2> (180-DEGREES)
                         | <integer: 3> (270-DEGREES)

<enumerated: L-line-progression> = <integer: 0> (90-DEGREES)
                                 | <integer: 1> (270-DEGREES)
```

8.5.X PEL ARRAY CLIP RECTANGLE

```
<PEL-ARRAY-CLIP-RECTANGLE-opcode: 3/4 3/5>
<integer: X1>
<integer: Y1>
<integer: X2>
<integer: Y2>
<point: first-corner>
<point: second-corner>
```

Page 47

Subclause 8.6: Add the following element representations:

8.6.X LINE TYPE DEFINITION

```
<LINE-TYPE-DEFINITION-opcode: 3/6 3/4>
<integer: line-type> <enumerated: dash-unit-selector>
<real: dash-repeat-length>
<enumerated: adaptive-flag>
<integer: list-of-dash-elements>*

<enumerated: dash-unit-selector> = <integer: 0> (VDC)
                                 | <integer: 1> (mm)
                                 | <integer: 1> (native device units)
                                 | <integer: 1> (abstract)

<enumerated: adaptive-flag> = <integer: 0> (NO)
                            | <integer: 1> (YES)
```

8.6.X HATCH STYLE DEFINITION

```
<HATCH-STYLE-DEFINITION-opcode: 3/6 3/5>
<integer: hatch-index>
<enumerated: style-indicator>
<enumerated: hatch-space-units-selector>
<real: DX-vector>
<real: DY-vector>
<real: duty-cycle-length>
<integer: list-of-hatch-elements>*

<enumerated: adaptive-flag> = <integer: 0> (parallel)
                            | <integer: 1> (cross-hatch)

<enumerated: hatch-space-units-selector> = <integer: 0> (VDC)
                                         | <integer: 1> (mm)
                                         | <integer: 1> (native device
                                                         units)
                                         | <integer: 1> (abstract)
```

8.6.X LINE CAP

```
<LINE-CAP-opcode: 3/6 3/6>
<enumerated: line-cap-indicator>

<enumerated: line-cap-indicator> = <integer: 0> (butt)
                                 | <integer: 1> (round)
                                 | <integer: 2> (projecting square)
```

8.6.X LINE JOIN

```
<LINE-JOIN-opcode: 3/6 3/7>
<enumerated: line-join-indicator>

<enumerated: line-join-indicator> = <integer: 0> (miter)
                                  | <integer: 1> (round)
                                  | <integer: 2> (bevel)
```

8.6.X EDGE CAP

```
<EDGE-CAP-opcode: 3/6 3/8>
<enumerated: edge-cap-indicator>

<enumerated: edge-cap-indicator> = <integer: 0> (butt)
                                 | <integer: 1> (round)
                                 | <integer: 2> (projecting square)
```

8.6.X EDGE JOIN

```
<EDGE-JOIN-opcode: 3/6 3/9>
<enumerated: edge-join-indicator>

<enumerated: edge-join-indicator> = <integer: 0> (miter)
                                  | <integer: 1> (round)
                                  | <integer: 2> (bevel)
```

8.6.X MITER LIMIT

<MITER-LIMIT-opcode: 3/6 3/10>
<real: miter-limit>

8.6.X EXTERNAL SYMBOL

<EXTERNAL-SYMBOL-opcode: 3/6 3/11>

<??????????>

ANSI X3H3

Information Processing Systems --

Computer Graphics --

Metafile for the Storage and Transfer

of Picture Description Information

Part 1

Functional Specification

(Clause 6)

Addendum 3

Draft Document 1.1

June 21, 1988

Clause 6: Add the following default specifications:

CONIC ARC TRANSFORMATION MATRIX:	identity matrix
PEL ARRAY CLIP RECTANGLE:	(0,0) upper left, (N-1,L-1) lower right, where N is the number of pels per line, and L is the number of lines in the last COMPRESSED PEL ARRAY element.
PEL ARRAY REFERENCE POINT:	upper left-hand corner point of the default VDC extent
Compound Line (BEGIN/END COMPOUND LINE)	No default
Compound Text Path (BEGIN/END COMPOUND TEXT PATH)	Right
Clipping Region (BEGIN/END CLIP REGION)	VDC Extent
Shielding Region (BEGIN/END SHIELD REGION)	VDC Extent
CLIP INDICATOR	On
SHIELD INDICATOR	Off
COLOUR REPRESENTATION METHOD	RGB

ANSI X3H3

Information Processing Systems --

Computer Graphics --

Metafile for the Storage and Transfer
of Picture Description Information

Part 3

Binary Encoding

Addendum 3

Draft Document 1.1

June 21, 1988

Subclause 7.3: Add the following notes (on Table 4):

nn FONTMETRIC DEFINITION: has 6 parameters:

P1: (integer) font index
P2: (xxxxx) character index
P3: (xxxxx) left bearing
P4: (xxxxx) right bearing
P5: (xxxxx) character height
P6: (xxxxx) baseline offset

nn CHARACTER KERNING MODE: has 1 parameter:

P1: (enumerated) character kerning mode: valid values are:

0 NONE
1 PAIR
2 SECTORED

nn CHARACTER KERNING TABLE: format to be determioed.

Px: (xxxxx) to be determined

nn FCS TYPE: has 1 parameter:

P1: (enumerated).FCS type: valid values are:

0 INTEGER
1 REAL
2 VDC

nn FCS INTEGER PRECISION: has 1 parameter:

P1: (integer) FCS integer precision: 8, 16, 24, or 32 are the only valid values.

nn FCS REAL PRECISION: has 3 parameters:

P1: (enumerated) form of representation for real values: valid values are:

0 floating point form
1 fixed point form

P2: (integer) field width for exponent or whole part (including 1 bit for sign)
P3: (integer) field width for fraction or fractional part

Legal combinations of values are:

P1	P2	P3	Result
0	9	23	32-bit floating point
0	12	52	64-bit floating point
1	16	16	32-bit fixed point
1	32	32	64-bit fixed point

nn FCS EXTENT: has 2 parameters:

P1: (fcspoint) first corner
P2: (fcspoint) second corner

If FCS TYPE is real, default FCS EXTENT is (0.0, 0.0), (0.9999...,0.9999...).

If FCS TYPE is INTEGER, default FCS TYPE is (0,0), (32767, 32767).

nn CAP HEIGTH: Parameterization to be determined.

***** Additional work required *****

Page 26

Subclause 7.5: Add the following notes (on Table 6):

nn BEGIN COMPOUND LINE: has no parameters.

nn END COMPOUND LINE: has no parameters.

nn BEGIN COMPOUND TEXT PATH: has no parameters.

nn END COMPOUND TEXT PATH: has no parameters.

nn BEGIN CLIP REGION: has no parameters.

nn END CLIP REGION: has no parameters.

nn BEGIN SHIELD REGION: has no parameters.

nn END SHILD REGION: has no parameters.

nn SHIELDING INDICATOR: hs 1 parameter:

P1: (enumerated) shielding indicator: valid values are:

0	OFF
1	ON

Page 28

Subclause 7.6: Add the following notes (on Table 7):

nn CONIC ARC: has 8 parameters:

P1: (point) start point
P2: (point) end point
P3: (real) first param
P4: (real) second param
P5: (real) third param
P6: (real) fourth param
P7: (real) fifth param
P8: (real) sixth param

nn CONIC ARC TRANSFORMATION: has 4 parameters:

P1: (VDC) R11
P2: (VDC) R12
P3: (VDC) R21
P4: (VDC) R22

nn PARAMETRIC SPLINE CURVE: has 6 parameters:

P1: (integer) curve type
P2: (integer) H degree of continuity
P3: (integer) N number of segments
P4: (??????????)
P5: (??????????)
P6: (??????????)

nn RATIONAL B-SPLINE CURVE: has 10 parameters:

P1: (integer) upper index of sums
P2: (integer) M degree of the basis function
P3: (enumerated) curve open flag: valid values are:

0 OPEN
1 CLOSED

P4: (enumerated) equation type flag: valid values are:

0 RATIONAL
1 POLYNOMIAL

P5: (enumerated) periodic flag: valid values are:

0 NON PERIODIC
1 PERIODIC

P6: (???????????)
P7: (???????????)
P8: (???????????)
P9: (real) start param
P10: (real) end param

nn COMPRESSED PEL ARRAY: has 8 parameters:

P1: (enumerated) T encoding: valid values are:

0 T4
1 T6

P2: (enumerated) P pel path: valid values are:

0 0 DEGREES
1 90 DEGREES
2 180 DEGREES
3 270 DEGREES

P3: (enumerated) L line progression: valid values are:

0 90 DEGREES
1 270 DEGREES

P4: (real) S pel spacing
P5: (real) spacing ratio
P6: (integer) N number of pels per line
P7: (integer) NL number of lines
P8: (xxxxxxx) pel array

nn PEL ARRAY CLIP RECTANGLE: has 6 parameters:

P1: (integer) X1
P2: (integer) Y1
P3: (integer) X2
P4: (integer) Y2
P5: (point) first corner
P6: (point) second corner

Page 32

Subclause 7.7: Add the following notes (on Table 8):

nn LINE TYPE DEFINITION: has 5 parameters:

P1: (integer) line type
P2: (enumerated) dash unit selector: valid values are:

0 VDC
1 MM
2 NATIVE DEVICE UNITS
3 ABSTRACT

P3: (real) dash repeat length
P4: (enumerated) adaptive flag: valid values are:

0 NO
1 YES

P5: (integer) list of dash elements

nn HATCH STYLE DEFINITION: has 7 parameters:

P1: (integer) hatch index
P2: (enumerated) style indicator: valid values are:

0 PARALLEL
1 CROSS HATCH

P3: (enumerated) hatch space units selector: valid values are:

0 VDC
1 MM
2 NATIVE DEVICE UNITS
3 ABSTRACT

P4: (real) DX vector
P5: (real) DY vector
P6: (real) duty cycle length
P7: (integer) list of hatch elements

nn LINE CAP: has 1 parameter:

P1: (enumerated) line cap indicator: valid values are:

0	BUTT
1	ROUND
2	PROJECTING SQUARE

nn LINE JOIN: has 1 parameter:

P1: (enumerated) line join indicator: valid values are:

0	MITER
1	ROUND
2	BEVEL

nn EDGE CAP: has 1 parameter:

P1: (enumerated) edge cap indicator: valid values are:

0	MITER
1	ROUND
2	BEVEL

nn EDGE JOIN: has 1 parameter:

P1: (enumerated) edge join indicator: valid values are:

0	MITER
1	ROUND
2	BEVEL

nn MITER LIMIT: has 1 parameter:

P1: (real) miter limit

nn EXTERNAL SYMBOL: has 1 parameter:

P1: (??????????)

ANSI X3H3

Information Processing Systems --

Computer Graphics --

Metafile for the Storage and Transfer
of Picture Description Information

Part 4

Clear Text Encoding

Addendum 3

Draft Document 1.1

June 21, 1988

Page 11

Subclause 5.4.1: Add the following words to the deleted words list:

CURVE
MATRIX

Page 11

Subclause 5.4.3: Add the following words to the unabbreviated words list:

CAP
CONIC
EXTERNAL
JOIN
LIMIT
MITER
PEL
REGION
SHIELD
SPLINE
SYMBOL

Page 12

Subclause 5.4.4: Add the following abbreviations:

COMPRESSED	CMPRSD
COMPOUND	CMPD
DEFINITION	DEF
KERNING	KERN
RATIONAL	RATNL
TRANSFORMATION	TRAN

Page 13

Subclause 5.4.5: Add the following derived element names:

Metafile Name	Element Name	Notes
BEGIN COMPOUND LINE	BEGCMPDLINE	
END COMPOUND LINE	ENDCMPDLINE	
BEGIN COMPOUND TEXT PATH	BEGCMPDTEXTPATH	
END COMPOUND TEXT PATH	ENDCMPDTEXTPATH	
BEGIN CLIP REGION	BEGCLIPREGION	
END CLIP REGION	ENDCLIPREGION	
BEGIN SHIELD REGION	BEGSHIELDREGION	
END SHIELD REGION	ENDSHIELDREGION	
SHIELDING INDICATOR	SHIELD	
FONTMETRIC DEFINITION	FONTMETRICDEF	
CHARACTER KERNING MODE	CHARKERNMODE	
CHARACTER KERNING TABLE	CHARKERNTABLE	
FCS TYPE	FCSTYPE	
FCS INTEGER PRECISION	FCSINTEGERPREC	
FCS REAL PRECISION	FCSREALPREC	
FCS EXTENT	FCSEXT	

CAP HEIGHT	TBD
LINE TYPE DEFINITION	LINETYPEDEF
HATCH STYLE DEFINITION	HATCHSTYLEDEF
LINE CAP	LINECAP
LINE JOIN	LINEJOIN
EDGE CAP	EDGECAP
EDGE JOIN	EDGEJOIN
MITER LIMIT	MITERLIMIT
EXTERNAL SYMBOL	EXTERNALSYMBOL
CONIC ARC	CONICARC
CONIC ARC TRANSFORMATION MATRIX	CONICARCTRAN
PARAMETRIC SPLINE CURVE	PARAMETRICSPLINE
RATIONAL B-SPLINE CURVE	RATNLBSPLINE
COMPRESSED PEL ARRAY	CMPRSDPELARRAY
PEL ARRAY CLIP RECTANGLE	PELARRAYCLIPRECT

Page 15

Subclause 6.2: Add the following Metafile Descripter element encodings:

```
FONTMETRIC DEFINITION          ::=   FONTMETRICDEF
                                        <I:FONTINDEX>   (positive)

                                     (format to be determined)

                                     <TERM>

CHARACTER KERNING MODE         ::=   CHARKERNMODE
                                        <SOFTSEP>
                                        <NONE|PAIR|SECTORED>
                                     <TERM>


CHARACTER KERNING TABLE        ::=   CHARKERNTABLE

                                     (format to be determined)

                                     <TERM>

FCS TYPE                       ::=   FCSTYPE
                                        <SOFTSEP>
                                        <INTEGER|REAL|VDC>
                                     <TERM>

FCS INTEGER PRECISION          ::=   FCSINTEGERPREC
                                        <SOFTSEP>
                                        <I:MININT>
                                        <SEP>
                                        <I:MAXINT>
                                     <TERM>

FCS REAL PRECISION             ::=   FCSREALPREC
```

```
                                        <SOFTSEP>
                                        <F:MINREAL>
                                        <SEP>
                                        <F:MAXREAL>
                                        <SEP>
                                        <I:DIGITS>
                                    <TERM>

FCS EXTENT                  ::=     FCSEXT
                                        <SOFTSEP>
                                        <P:FIRSTCORNER>
                                        <SEP>
                                        <P:SECONDCORNER>
                                    <TERM>

CAP HEIGHT                  ::=     TBD<TERM>
```

Page 18

Subclause 6.5: Add the following Control element encodings:

```
BEGIN COMPOUND LINE         ::=     BEGCMPDLINE<TERM>

END COMPOUND LINE           ::=     ENDCMPDLINE<TERM>

BEGIN COMPOUND TEXT PATH    ::=     BEGCMPDTEXTPATH<TERM>

END COMPOUND TEXT PATH      ::=     ENDCMPDTEXTPATH<TERM>

BEGIN CLIP REGION           ::=     BEGCLIPREGION<TERM>

END CLIP REGION             ::=     ENDCLIPREGION<TERM>

BEGIN SHIELD REGION         ::=     BEGSHIELDREGION<TERM>

END SHIELD REGION           ::=     ENDSHIELDREGION<TERM>

SHIELDING INDICATOR         ::=     SHIELD
                                        <SOFTSEP>
                                        <OFF|ON>
                                    <TERM>
```

Page 19

Subclause 6.6: Add the following Graphical Primitive element encodings:

```
CONIC ARC                   ::=     CONICARC
                                        <SOFTSEP>
                                        <P:STARTPOINT>
                                        <SEP>
                                        <P:ENDPOINT>
                                        <SEP>
                                        <F:FIRSTPARAM>
                                        <SEP>
                                        <F:SECONDPARAM>
                                        <SEP>
```

```
                                        <F:THIRDPARAM>
                                        <SEP>
                                        <F:FOURTHPARAM>
                                        <SEP>
                                        <F:FIFTHPARAM>
                                        <SEP>
                                        <F:SIXTHPARAM>
                                      <TERM>

CONIC ARC TRANSFORMATION MATRIX  ::=  CONICARCTRAN
                                        <SOFTSEP>
                                        <VDC:R11>
                                        <SEP>
                                        <VDC:R12>
                                        <SEP>
                                        <VDC:R21>
                                        <SEP>
                                        <VDC:R22>
                                        <SEP>
                                        <P:COORDOFFSET>
                                      <TERM>

PARAMETRIC SPLINE CURVE          ::=  PARAMETRICSPLINE
                                        <SOFTSEP>
                                        <I:CURVETYPE>
                                        <SEP>
                                        <I:CONTINUITYDEGREE>
                                        <SEP>
                                        <I:NUMSEGMENTS>
                                        <SEP>

                                        (??????????)

                                      <TERM>

RATIONAL B-SPLINE CURVE          ::=  RATNLBSPLINE
                                        <SOFTSEP>
                                        <I:UPPERINDEXOFSUM>
                                        <SEP>
                                        <OPEN|CLOSED>
                                        <SEP>
                                        <RATIONAL|POLYNOMIAL>
                                        <SEP>
                                        <NONPERIODIC|PERIODIC>
                                        <SEP>

                                        (???????????)

                                        <SEP>
                                        <F:STARTPARAM>
                                        <SEP>
                                        <F:ENDPARAM>
                                      <TERM>

COMPRESSED PEL ARRAY             ::=  CMPRSDPELARRAY
                                        <SOFTSEP>
                                        <T4|T6>
                                        <SEP>
                                        <0|90|180|270>
```

```
                                        <SEP>
                                        <90|270>
                                        <SEP>
                                        <F:PELSPACING>
                                        <SEP>
                                        <F:SPACINGRATIO>
                                        <SEP>
                                        <I:PELSPERLINE>
                                        <SEP>
                                        <I:NUMBEROFLINES>
                                        <SEP>

                                        (?????????????)

                                    <TERM>

PEL ARRAY CLIP RECTANGLE        ::=  PELARRAYCLIPRECT
                                        <SOFTSEP>
                                        <I:X1>
                                        <SEP>
                                        <I:Y1>
                                        <SEP>
                                        <I:X2>
                                        <SEP>
                                        <I:Y2>
                                        <SEP>
                                        <P:FIRSTCORNER>
                                        <SEP>
                                        <P:SECONDCORNER>
                                    <TERM>
```

Page 24

Subclause 6.7: Add the following Attribute element encodings:

```
LINE TYPE DEFINITION            ::=  LINETYPEDEF
                                        <SOFTSEP>
                                        <I:LINETYPE>
                                        <SEP>
                                        <VDC|MM|NDU|ABST>
                                        <SEP>
                                        <F:DASHREPEATLEN>
                                        <SEP>
                                        <NO|YES>
                                        <SEP>

                                        (????????)

                                    <TERM>

HATCH STYLE DEFINITION          ::=  HATCHSTYLEDEF
                                        <SOFTSEP>
                                        <I:HATCHINDEX>
                                        <SEP>
                                        <PARALLEL|CROSSHATCH>
                                        <SEP>
                                        <VDC|MM|NDU|ABST>
                                        <SEP>
```

```
                                        <F:DX>
                                        <SEP>
                                        <F:DY>
                                        <SEP>
                                        <F:DUTYCYCLE>

                                        (!!!!!!!!!!!!!!!)

                                    <TERM>

LINE CAP                    ::=     LINECAP
                                        <SOFTSEP>
                                        <BUTT|ROUND|SQUARE>
                                    <TERM>

LINE JOIN                   ::=     LINEJOIN
                                        <SOFTSEP>
                                        <MITER|ROUND|BEVEL>
                                    <TERM>

EDGE CAP                    ::=     EDGECAP
                                        <SOFTSEP>
                                        <BUTT|ROUND|SQUARE>
                                    <TERM>

EDGE JOIN                   ::=     EDGEJOIN
                                        <SOFTSEP>
                                        <MITER|ROUND|BEVEL>
                                    <TERM>

MITER LIMIT                 ::=     MITERLIMIT
                                        <SOFTSEP>
                                        <F:MITERLIMIT>
                                    <TERM>

EXTERNAL SYMBOL             ::=     EXTERNALSYMBOL

                                    (to be determined)

                                    <TERM>
```

This page left intentionally blank.

APPENDIX 8

METAFILE REFERENCE MODEL AND POSITIONS

This page left intentionally blank.

A Reference Model to Aid in Understanding the Role of Multiple Metafile Types

U.S. Position

17 May 1988

I. Motivation

Considering the current set of graphics standards completed and in progress, and the current set of proposals for Addenda to IS 8632, there appears to be a pressing need for a useful reference model with which to understand the different varieties of metafile and how and where they fit into graphics systems.

The relative newness of the Addendum process within SC24 has led us to a situation where there is much confusion and lack of clarity with respect to the scope and goals of the CGM Addenda.

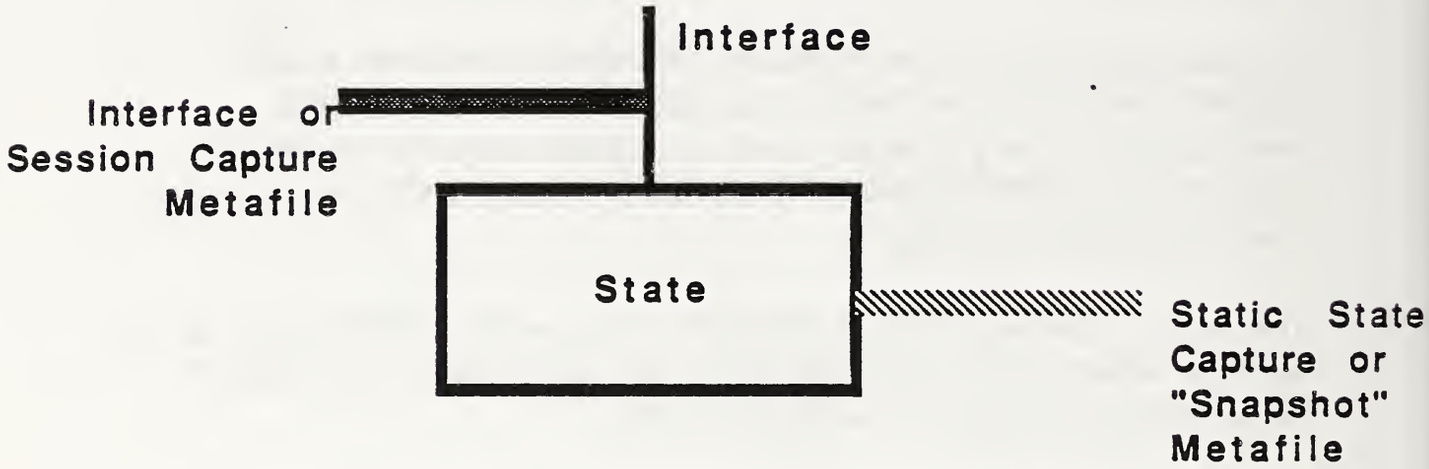
We hope that this contribution will prove to be a useful reference model for re-evaluating the Addenda in progress, for aligning their functionality with the appropriate standards, and as an input for the Reference Models group for future work.

II. Model and Definitions

A. Types of Metafiles. We believe it is useful to distinguish between two fundamentally different types of metafile (see Diagram 1):

Interface or Session Capture - this type of metafile is intimately bound to the semantics of a particular interface, as defined in a semantic standard. This kind of metafile is frequently referred to as an audit trail. For example, the GKSM is a session capture metafile which captures the information flowing across the workstation interface defined by GKS.

Static State Capture - this type of metafile is, in effect, a "snapshot". For example, the PHIGS archive file is a snapshot of the state of zero or more PHIGS structures; the CGM is a snapshot of a picture definition. The state capture variety of metafile need not be as tightly bound to the semantics of the generator of the metafile as the interface capture



Basic Building Blocks for Metafile Reference Model

Diagram 1

variety, although it may be. For example, the CGM defines "pictures" which can be viewed as "things unto themselves", comprising sufficient information that they may be generated or interpreted by a variety of different graphics systems not limited to those based on API's defined by SC24 graphics standards. Conversely, the PHIGS archive file contains information tightly bound to the semantics of the PHIGS standard, and would not be expected to be used by clients other than PHIGS implementations.

B. Reference Model. Diagram 2 illustrates the overall metafile reference model, showing all possible locations of both types of metafiles (session capture and state capture) within the context of the current layered or pipeline model used in this generation of SC24 standards. Not all of these metafiles need be defined as standards. Also, not all graphics levels (boxes), interfaces (connectors), or metafiles need be explicitly present in an implementation, which can be viewed as a degenerate case of the overall reference model. Any or all of the interfaces, metafiles, or processing environments may be replaced by equivalent facilities based on proprietary definitions or *de facto* standards, without invalidating the overall reference model.

Each of these metafiles has a distinct set of semantics, and a distinct purpose and use. We believe that the graphics standards community, and the graphics industry, are best served by cleanly defined standards which serve exactly one of these purposes as represented by a single point in the model.

Diagram 3 illustrates how this basic reference model could be applied to a layered standards-based graphics system. Note, however, that it is only one possible arrangement.

C. Harmonization of Multiple Metafile Standards. We believe that separate metafile standards can be harmonized through use of common encoding techniques, deliberately choosing to encode identically those elements which are semantically identical in these standards.

A useful analogy can be drawn to language bindings. For any semantic standard, language bindings can be generated for several languages. For any language, bindings can be made to several semantic standards. Although this could conceivably have led to a wild proliferation of binding techniques and inconsistent choices, the chaos has been managed quite well by means of a central "generic issues" log which is applied to all language bindings, and by a commitment to harmonize closely related bindings through conscious application of consistent binding techniques,

Metafile Reference Model

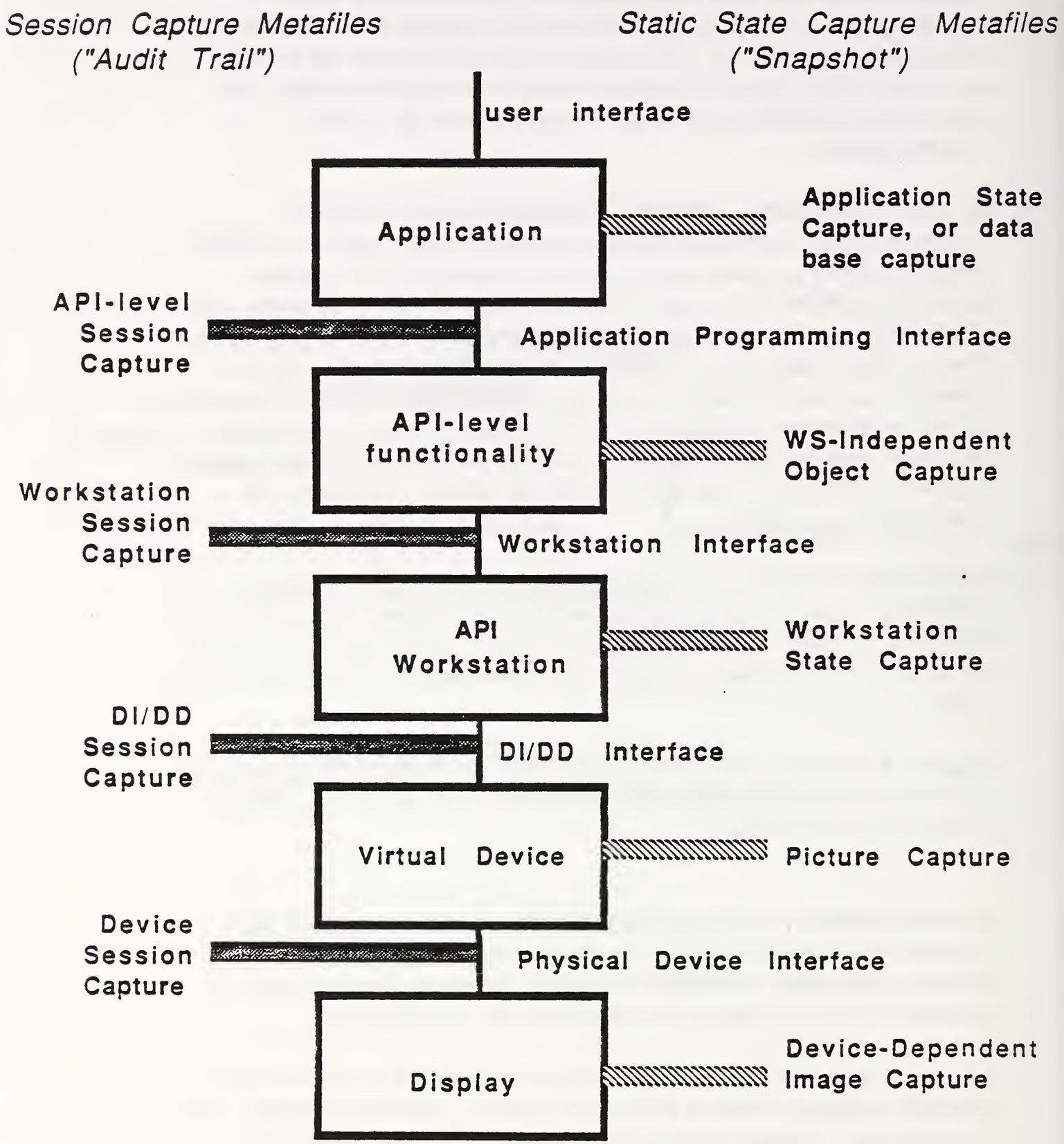
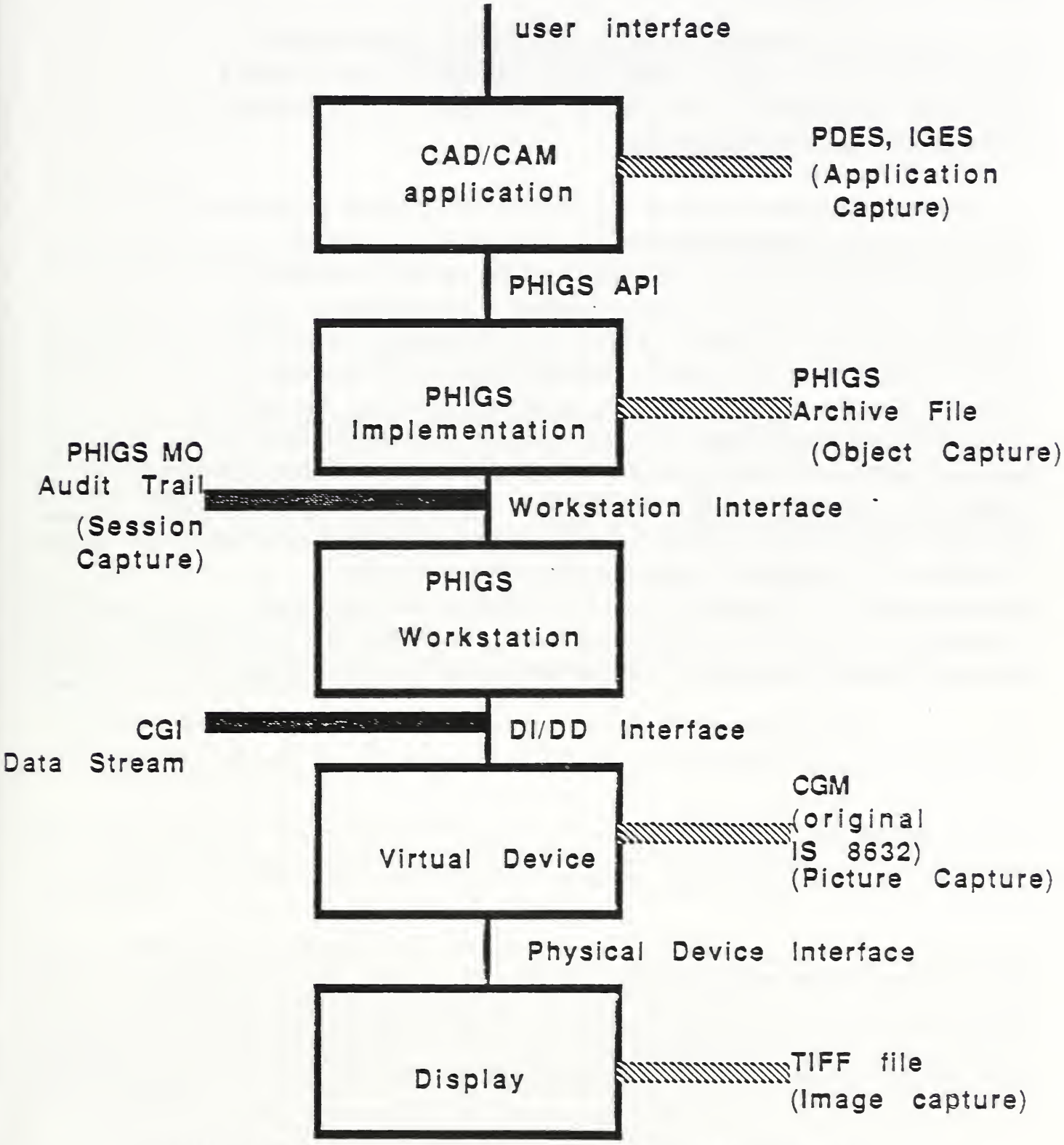


Diagram 2

An Instance of a Layered Architecture Based on the Reference Model

(Diagram 3)



NOTE: there is no implication that PHIGS need be built on top of CGI, nor that CGM be generated through CGI. This is only one possible arrangement, for illustration only.

naming and abbreviation conventions, and deliberate reuse of components.

Towards this end, we encourage the WG3 Metafile Rapporteur Group to produce documents which describe the encoding techniques developed so far, and which provide guidance in how to apply these encoding techniques to generate compatible encodings for additional metafile standards.

D. The question of multiple usage. It is not always the case that information from a metafile re-enters the same system from whence it came, nor is it necessarily the case that it re-enters the system at the same point where it was generated.

It should be recognized, however, that when either of these situations occurs, additional software processing will be needed to map the semantics inherent in the metafile type onto the semantics available for the type of metafile which can be processed by the interpreter. This mapping might be in the form of a filter which turns one form of state capture metafile into another (e.g., taking a PHIGS archive file and creating CGM files for picture output), or an emulator which can take a session capture metafile and "play it out" over a different interface (e.g., taking a GKSM and feeding it into a PHIGS workstation). This additional processing may also happen within a processing environment; for example, a PHIGS implementation might support importing CGM picture capture metafiles, and provide the necessary interpretation to insert corresponding PHIGS elements into the PHIGS structure store (as described in Annex H of the PHIGS standard). More than one such mapping can occur for any given pair of metafile and consumer of the metafile.

U.S. Position on Structuring and Progression of CGM Addenda

I. Introduction

The U.S. has submitted a paper entitled "A Reference Model to Aid in Understanding the Role of Multiple Metafile Types", document X3H3/88-77. These comments are made in the context of that model.

The addendum process was chosen as the fastest available method for producing a standardized GKSM based on the CGM. However, a consideration of the three addenda in process reveals that the lack of a NWI ballot to agree upon scope and goals has created more problems than it has solved.

The U.S. believes it is inappropriate to use the addendum process to alter the CGM standard in ways that change the metafile type (static state capture) or its location in the graphics pipeline. The addendum process should be reserved for those additions which enhance the functionality in ways consistent with the original scope, goals, and architecture of the standard. The U.S. believes that the addendum process should not be used as a general means of building quite different graphics standards under the guise of a modification to an existing standard.

II. Consideration of Addendum 1

The Metafile Reference Model helps clarify the relationship of addendum 1 to ISO 8632. There are at least two goals in addendum 1 (there may be more):

1. provide a GKSM (workstation session capture metafile) to replace annex E of GKS;
2. provide more advanced static picture capture capabilities (the additional functionality taken from CGI, with some restrictions);

The Metafile Reference Model shows the first to be a distinct type of metafile from ISO 8632, whereas the second is really a proper extension of ISO 8632. Accordingly, it is inappropriate to treat the first as an addendum to ISO 8632; rather it would best be progressed by one of the other methods mentioned in X3H3/88-78.

The US acknowledges the advanced state of addendum 1 and the desirability of not incurring further delay in progressing the GKSM content of addendum 1. However if it were concluded that the work could be restructured in a way that is consistent with the Metafile Reference Model, without incurring further delay in standardizing the GKSM content of addendum 1, the US would support such a change. We believe it is within the authority of SC24, for example, to convert the work in progress from an addendum to CGM to an addendum for a normative annex to GKS, without causing a setback to the desired schedule.

III. Consideration of Addendum 2

Please see document X3H3/88-78, "U.S. Comments on SC24 N23: Working Draft for ISO 8632 3D Addendum (CGM Addendum 2)".

IV. Consideration of Proposed Addendum 3

The proposed addendum 3 is intended to be a metafile that is of type static picture capture (like CGM) with extended functional capabilities defined for use by both both standard and non-standard client systems. Thus, this work would correctly be undertaken and progressed as an addendum to ISO 8632. The US supports progressing this work as an addendum to CGM (ISO 8632).

V. Conclusion

The U.S. believes it to be desirable to restructure the work in progress, such that only that functionality which is consistent with the original scope, goals, and architecture of IS 8632 be progressed as addenda to that standard. Work which results in a different type of metafile should be progressed as a separate standard, or as a normative annex to the functional standard to which the metafile relates (e.g. GKS, GKS-3D, PHIGS).

However, much benefit can be realized by utilizing the encoding techniques, and as far as possible the exact element encodings, already standardized in IS 8632 parts 2 through 4. This will serve to harmonize the different metafiles, without requiring them all to be formulated as addenda to IS 8632.

U.S. Comments on SC24 N23: Working Draft for ISO 8632 3D Addendum (CGM Addendum 2)

I. Introduction

The US has two fundamental concerns with SC24 N23, the proposed 3D addendum to CGM:

1. the scope and goals of the addendum are unclear, and apparently contrary to decisions taken at the May 1987 SC21/WG2 meeting;
2. the addendum process may not be the appropriate method for realizing the metafile requirements of constituencies identified for addendum 2 (as well as addendum 1).

As part of its review of both addendum 2 and addendum 1, the US has derived a reference model for metafiles (see document X3H3/88-77). We believe that this model clearly illustrates the problems with CGM addendum 2 (as well as the other proposed addenda). We have included a description of that model with these comments, and we comment on addendum 2 in the context of the model.

II. Addendum 2 Comments: Issues of Scope and Purpose

The attached Metafile Reference Model (MRM) shows that there are two fundamental types of metafiles (session capture and static state capture) and many places in a typical graphics architecture that a metafile may be captured; therefore, there are many useful metafiles. In the MRM, the CGM (ISO 8632) is a static state capture metafile at the virtual device level – it is a picture capture metafile.

In Valbonne, it was decided that the scope of the CGM addendum 2 is to provide a replacement for the (non-normative) annex E of GKS-3D. The exact scope and purpose of addendum 2, as presented in N23, are unclear. We identify at least 3 metafiles in the MRM that addendum 2 purports to support:

1. GKSM-3D
2. PHIGS Archive File
3. PHIGS Metafile for Workstation MO

The latter two metafiles have not been previously identified as part of the scope of addendum 2, nor have the requirements for them been carefully studied by the metafile working group. Furthermore work on the PHIGS archive is already in progress within ISO and has now reached the DIS stage.

The US believes that addendum 2 should be analyzed, restructured and redefined in the context of the MRM. A set of target metafiles should be identified and the scope of each metafile clearly defined. Work overlapping other active ISO projects should be discontinued or consolidated with the other work.

III. Addendum 2 Comments: Issues of Organization of Work

The US believes that it is inappropriate to use an addendum to CGM to progress standardization of 3D metafiles, since none of the three types of files (listed above) which could be specified by this addendum is at the same level (Virtual Device) and of the same type (static picture capture) as the CGM (ISO 8632). To standardize metafiles that are not properly extensions of ISO 8632 we recommend one of three methods:

1. as a part of or addendum to another standard, which they are designed to serve;
2. as a distinct independent standard;
3. as a new part of an existing metafile standard.

ISO metafile experts should play a major role under any of the options, and already developed element sets and encoding techniques should be reused to the maximum extent possible.

APPENDIX 9

TUCSON METAFILE REPORT

This page left intentionally blank.

Delegation Report
WG3 Metafile Rapporteur Group at the Tucson Meeting of ISO/IEC JTC1/TC97/SC24, June 1988

Lofton Henderson, Head of Sub-delegation

1. Overview

In early 1988 CGM Addendum 1 was circulated for PDAD ballot, and CGM Addendum 2 Working Draft was circulated for comment. In April SC24 had a Special Working Group (SWG) meeting which recommended revision and improvement of the way of progressing graphics standards, and urged reference model and user requirements work precede new projects.

The Tucson meeting was an editing meeting for the Metafile Rapporteur Group of WG3. The main tasks were to advance the status of CGM Addendum 1 and Addendum 2. For Addendum 1, the results of the PDAD ballot were pre-processed at Blakeney, England in April and the goal of this meeting was to complete that processing and prepare the DAD text. For Addendum 2, the goal was to process the comments on the Working Draft and prepare PDAD text.

At the Fairfax meeting of X3H3, a Metafile Reference Model was derived and submitted as input to the Tucson meeting. The impetus for this work came from both the recommendations of the SWG and internal dissatisfaction within X3H3 over the way the addendum process was being used for CGM.

U.S. positions were generated recommending splitting Addendum 1 into a static picture capture portion and a GKS audit trail portion. The first would continue as an addendum to ISO 8632 CGM, and the second as a normative annex to GKS. For Addendum 2 it was proposed that 3D metafiles not be progressed as addenda to ISO 8632 but as parts or annexes of their respective 3D standards.

There was substantial agreement with these positions at Tucson. A number of other national bodies had simultaneously arrived at similar positions. Consequently, the CGM addenda are being split, restructured, and progressed substantially along the lines recommended in the U.S. position papers (X3H3/88-77, 78, 79).

2. Meeting Schedule

— 28 June, Metafile Rapporteur Group convenes.

— 1 July, Metafile Rapporteur Group adjournes.

— 2 July, liaison meeting with SC18 (no R.G. activity).

— 3-6 July, drafting and document markup.*

— 6 July, liaison meeting with CGI.

— 7 July, WG3 plenary.

— 8-9 July, SC24 plenary.

* Also took place during R.G. meeting, and during plenaries.

3. Attendance

Attendance included:

— US: Lofton Henderson, Andrea Frankel, Barbara Lurvey, Peter Bono;
— UK: Anne Mumford (Addendum 1 document editor), Alan Francis;
— France: Bernard Troucherie;
— Germany: Eckhard Moeller (rapporteur), Peter Egloff;
— Austria: Emaneul Wenger;
— Denmark: Kurt Alstrup;

— Japan: Ichiron Nishioka

4. Administrative Notes

Eckhard Moeller, the rapporteur of the Metafile Rapporteur Group within WG3, will not be continuing due to a change of job responsibility. He will continue through the next R.G. meeting, which will be an editing meeting to process the results of the next ballots on Addendum 1 and Addendum 2, in June-July 1989.

The CGM Maintenance Rapporteur Group had been allowed to lapse, due to an oversight. There is a growing list of corrections to be made to CGM. Alan Francis was reappointed as rapporteur for the Metafile Maintenance Rapporteur. At one time there was a Metafile Maintenance R.G., but it lapsed. At that time Andrea Frankel and Lofton Henderson were the R.G. members for the U.S. It is not clear what if anything need be done to reestablish membership in the group.

5. Technical Action List for the Meeting

The following comprised the major technical activities of the meeting.

1. Discuss US Metafile Reference Model (MRM), similar UK positions, and other national positions; split and restructure both CGM addenda accordingly, into two CGM addenda and annexes to GKS(3D).
2. Addendum 1 technical issues resolution from PDAD ballot and disposition of comments.
3. Drafting and document markup to effect the splitting of Addendum 1 and Addendum 2 (both still in progress).
4. Liaison with GKS R.G. on GKS requirements for metafiles, as well as on progressing of GKSM content of Addendum 1 as a normative annex of GKS.
5. Liaison with VTR on conformance topics of extended CGM and GKSM, as well as registration topics.
6. Resolutions, including position on "addendum 3", etc.
7. CGI liaison on areas of overlapping functionality.
8. Liaison with SC18 (not an R.G. activity per se).
9. Schedule and Future Work

These list items are dealt with in the following sub-sections.

5.1 Reference Models and Structure of Work

5.1.1 The MRM After introductory remarks, introduction of delegates, discussion and adjustment of the agenda, the U.S. presented Metafile Reference Model (MRM) which was developed at Fairfax (X3H3/88-77). The MRM recognized that there are, in a typical graphics architecture, very many different types of metafile that could be defined and standardized. These be divided roughly into two classes: static (state capture) and dynamic (session or interface capture). Within each class there can be metafiles defined at many different levels.

According to this model, the CGM (ISO 8632) is designated as a static picture capture metafile at roughly the Virtual Device level. The two ISO addenda throw many features into CGM which confuse both its static/dynamic nature and the level at which it exists in the graphics architecture. The U.S. model and associated positions recommend clearly identifying the goal metafiles (based on user requirements), splitting the work in progress, and progressing the work as addenda to CGM (for static picture metafiles) or additions to other standards (for dynamic metafiles, e.g., GKSM, ..).

5.1.2 National Responses to the MRM The R.G. discussed at length the model and its implications on what metafiles to standardize and how to progress the work. The R.G. was unanimous in endorsing the model and the principle of splitting and restructuring the work. The U.K. had been dissatisfied with the CGM addenda on philosophical grounds, and was particularly supportive of the principles of the model

and the idea of restructuring.

5.1.3 Splitting the Work — Conceptual There was no objection to continuing to progress the static portion of Addendum 1 as a CGM addendum, to form a "static structured picture capture" metafile.

Most of the problems and discussion were related to what to do about the GKSM content of Addendum 1. From a practical standpoint, two categories of concerns were raised:

1. the restructuring should not slow down the GKSM (France);
2. is a standard GKSM (similar to annex E of GKS) needed at all; is there a more useful dynamic metafile (U.K., Germany, Austria).

The U.S. proposed making the GKSM a normative annex to GKS, to replace the current annex E. To avoid delay, we recommended doing this under the authority of the Egham resolutions, as a simple conversion of that mandated work from a CGM addendum to a GKS addendum. We further recommended that a New Work Item (NWI) was not required, and that the work should be progressed at the same pace as Addendum 1 and be done by the metafile experts in the Metafile R.G.

Although there was general agreement that GKSM belonged in GKS, there was significant debate on where and how. There was significant disagreement from the U.K. on the proposition that the GKSM could simply be attached to GKS — within the U.K. delegation (though not necessarily the R.G. members) there was the belief that SC24 would not allow the GKS annex without a NWI.

Further confusion arose concerning due to the uncertain scope and timetable of GKS revision. Would it be a fast minimal maintenance effort to fix problems in GKS 85? Or would it be a more general revision to produce the new API, GKS 9x? Should GKSM be an ammendment to GKS 85 or should it track what was happening in maintenance and revision to the API? The Metafile R.G. consensus was that the fastest possible GKSM addition to GKS 85 was the only realistic way that progress could be made.

The biggest issue, however, was not how to progress GKSM but whether a standard GKSM audit metafile was needed at all. The U.K. and Austria, and to some extent Germany, maintained that a standard GKSM was not needed. Germany proposed, and there was agreement from some of the other delegates, that a "dynamic picture capture" metafile was much more useful than the pure GKSM audit trail. The dynamic picture metafile would allow multiple pictures, as opposed to the single picture of GKSM. All GKS control functions that explicitly commanded clearing or regeneration (CLEAR, UPDATE, REDRAW ALL SEGMENTS, ..) would be mapped to new pictures and would not be written to the metafile. Those functions with potentially dynamic effects (representation functions, workstation transformation, segment attributes and manipulation, ..) would be written to the metafile.

There was further discussion and yet more options on this topic were introduced during the WG1 GKS R.G. liaison meeting — see below.

The resolution (by meeting's end) was that we would progress the static structured metafile and the Egham-mandated GKSM audit trail. Dynamic picture metafiles could be undertaken later after the requirement is studied and demonstrated. (The extensibility is built into the ammended CGM to make it relatively easy).

5.1.4 Particular Technical Issues Arising from the Split Splitting the addenda reopened some issues. The split of the static and dynamic portions of Addendum 1 and the progression of the GKSM audit metafile as a part of GKS led to some technical issues and resolutions.

1. Issue: What should be done with Color Table and Pattern Table, which are potentially dynamic and are allowed in the picture body in CGM?

   Resolution: It was acknowledged that CGM IS 8632 was itself somewhat "impure" in regard to being static. Color Table and Pattern Table are allowed in the picture body. (The standard says that the effect of changing an index with primitive bound to it is not standardized.) CGM Add.1 will still allow the elements in the picture body, but will also allow them in the Picture Descriptor. Use in the picture body will be discouraged.

2. Issue: Should there be segments the static picture metafile at all? If the static picture metafile is at the level of the virtual device, and if it is static, then the MRM might imply that segments should not be in the metafile.

Resolution: Segments can be considered as a shorthand, for data compression, at this level of the MRM. Furthermore, they even exist in GKS at this level (the device level), as WDSS. Segments are useful and are not inconsistent with the MRM. However they must be static segments — the grammar of static metafiles will not allow segment control functions such as DELETE, and will not allow segment attribute functions to be used in a way that would imply picture dynamics. In static metafiles segment attributes are only allowed within segments, at the beginning, before any primitives. The CGI segment control functions that are eliminated from CGM Add.1 are DELETE SEGMENT, DELETE ALL SEGMENTS, RENAME SEGMENT, REOPEN SEGMENT, REDRAW SEGMENT, REDRAW ALL SEGMENTS.

A peripheral issue is whether only global segments should be allowed. There were arguments that this was the only consistent way (according to the MRM) to allow segments in the static picture metafile. It was decided is that local segments caused no problem, as long as dynamic effects were proscribed by the grammar as described above.

3. Issue: Many elements in Addendum 1 were taken from functions of CGI — segment control and attribute, bundle representation, workstation transformation, primitive attributes, etc. Is it appropriate that GKSM still be expressed in terms of these elements and the existing elements of CGM if GKSM is to be a GKS annex?

   Resolution: if the elements were still to be a part of static CGM, such as separate TEXT FONT INDEX and TEXT PRECISION elements, then they would be used in GKSM; if the elements would no longer exist in static CGM — the PREPARE VIEW SURFACE, MAKE PICTURE CURRENT, .. from which the GKSM CLEAR, REDRAW SEGMENTS, .. had been built — then they would be eliminated also from GKSM and the GKS functions CLEAR, UPDATE, REDRAW, .. would be encoded as GKSM elements.

   The basic principle is to maintain as much overlap as possible in metafile specifications, and only invent and add new elements where such were needed. Such are needed in cases where semantics differ significantly, or an awkward mapping is required to define a function.

   Based on the principles enumerated above, some of the issues about whether GKSM needs different elements and encodings for some specific functions are still being sorted out.

4. Issue: What do category and element set mean and what values should be standardized?

   Resolution: Better definition of METAFILE CATEGORY and METAFILE ELEMENT SET shorthands was begun at Blakeney and continued in Tucson subsequent to the restructuring of the addenda. Category now refers purely to grammar of the metafile, although a maximum element set is implied in some cases. The following categories are currently defined (these are probably not the final names):

   - basic_static — the current CGM;
   - structured_static — the grammar of the structured static CGM;
   - gksm_audit — the dynamic GKSM audit trail metafile;
   - 3d equivalents of the last two (more about 3d later).

   The following element set shorthands will be defined:

   - drawing and drawing_plus_control — the shorthands of the current CGM.
   - structured_static_all — all elements in the structured static CGM.
   - gksm_static_all — those elements that would exist in a GKSM static picture capture; special geometric primitives like circles and arcs are not included (see next);
   - extended_primitives_set — all those primitives in CGM plus Addendum 1 which are beyond the basic primitive set of GKS.
   - gksm_audit_all — those elements (less the extended primitives) that would be used in a GKSM audit trail metafile.
   - 3d equivalents of all of the above.

   The intention in splitting the special primitives out from the GKS element sets was to give implementations a convenient way to express both the case where GKS GDP maps to metafile GDP and the case where it maps to the CGM extended primitives.

5. Issue: what should happen to CGI pick related functions in the static metafile?

   Resolution: pick id and segment pick priority stay in; they are considered to be primitive attributes and segment attributes that record useful information in a snapshot. Segment detectability is eliminated.

6. Issue: Should Pixel Array and Drawing Mode remain in?

   Resolution: There were arguments that this was below the static picture level in the MRM. There was further concern over its (in)stability in CGI (it has been made a raster function, not treated in the pipeline as are geometric primitives). Counter arguments were that a metafile a one level could consistently incorporate functionality from a lower level (comparison to OSI was used to support this position). And the CGM grammar could take care of excluding it from segments and otherwise being treated as a geometric primitive. Pixel Array was retained in Addendum 1. Drawing Mode was similarly retained, but without the new values added by CGI at this meeting.

5.2 Addendum 1 PDAD Technical Issues Resolution

One of the output documents of the Tucson meeting (not yet received) will be a disposition of comments on the Addendum 1 PDAD ballot.

Much of the technical issue resolution done at Blakeney became moot after the splitting of CGM Add.1 and GKS Add.1. For example, many of the CGI control functions were removed, so issues related to those disappeared. Significant Blakeney results that remained intact include:

1. The METAFILE CATEGORY and METAFILE ELEMENT SET elements are made orthogonal as much as possible — category refers on to grammar (but may imply a maximum element set).
2. MODIFY FONT LIST and MODIFY CHARACTER SET LIST elements are removed.
3. The grammar for Addendum 1 will show a necessary order for some of the Metafile Descriptor elements (Version, Element Set, Category, Description), as per a U.S. comment.
4. The GKSM0 element set has been removed.

In addition, numerous changes were agreed to make the CGM elements match corresponding CGI functions. The description is being modified throughout to be more appropriate for a metafile specification — much of it was borrowed from functional standards (e.g., CGI) and had not been adapted to CGM.

5.3 Drafting to Split the Addenda

A significant amount of drafting for the CGM Add.1 was accomplished at Tucson. Much (most) of the drafting for GKSM remains to be done. A number of particular technical problems are being sorted out according to the principles we derived for how to accomplish the split.

5.4 Liaison with GKS

On the afternoon of 30 June there was a liaison meeting with the GKS maintenance group. CGM was seeking some guidance as to what types of metafiles were perceived as needed for GKS (refer above, to the doubts over whether a standard audit trail GKSM is needed). CGM also wanted guidance on procedural issues for attaching a GKSM annex to GKS. McConnell added to the agenda the topics of standard items types for GKS, and putting the additional primitives of the metafile into the GKS API.

Considerable time at the meeting was spent reviewing the current status and plans of metafile addenda. A new type of metafile of interest to some GKS constituents was proposed — an audit trail of the API.

Finally, there was consensus on two points:

— The Metafile R.G. should complete work on an audit trail GKSM, to be attached to GKS 85 as a normative annex (unanimous);
— The GKS Maintenance R.G. and the Metafile R.G. should continue to liaise to determine what other sorts of metafiles need be standardized for GKS.

5.5 Liaison with VTR

On 1 July there was a liaison meeting between Metafiles and VTR. Topics included:

— Conformance testing and formal grammars;
— The role of Application Profiles (APs);
— Current status of conformance testing for CGM;
— Registration procedures.

On the first point, it was agreed that conformance testing could be greatly enhanced if the formal grammars of the metafile standards were normative. (The formal grammars of the CGM addenda will be). It is generally agreed that APs provide the basis for testing generators and interpreters. In CGM Add.1 there will still be no specification of generator and interpreter behavior (although the formal grammar does give something to test the generator on). There is work going on in ISO on standardizing APs. This, it was generally agreed, would help conformance testing and would be a good idea. Pink and Mumford will look into this further.

There was a report on CGM testing work being done by GMD. A set of test metafiles is being developed, based on GKS. These are simple metafiles with just a single primitive or two. There is also work on a syntax analyzer, to check generator output. It is felt that it will be 1/2 to 1 year before the testing tools are ready (it is unclear what "ready" means here).

The Registration Procedures, TR 9973, are finished. The final draft is being sent to the secretariat this week. The proposals for additional linetypes and hatch styles, for engineering drawing support, have been mailed. Skall wants a 90 day ballot on these.

5.6 Liaison with CGI

Barbara Lurvey (WG3 Head of Delegation) met with and worked with the Metafile R.G. for its entire official 4-day meeting. Her efforts were invaluable in bringing CGI positions and issues into CGM, and taking CGM concerns back to CGI. It was hoped that topics of interest to CGM could be dealt with at the very beginning of the CGI meeting, with CGM experts present. The very heavy workload of CGI forced the liaison meeting to the end.

The major concern within CGM was over stability of those CGI functions which had been incorporated into Addendum 1. There was particular concern over Pixel Array and Drawing Mode, and the appearance of the new clipping mode functions. CGM intends to go to DAD ballot with Addendum 1, whereas CGI is going to 2nd DP. This creates a certain awkwardness since CGM is following CGI on the functionality and encoding of the set of functions in question. CGM desired a resolution that would delineate the area of overlap between the two standards and "freeze" that set in both standards (except for the fixing of serious errors).

CGI endorsed progressing CGM Add.1 to DAD, with inclusions of those CGI functions that still made sense in a static structured metafile. CGI recommended that Pixel Array and Drawing Mode be removed (see the discussion above). It was not until the WG3 plenary that it was resolved to leave them in.

The next editing meetings of CGM and CGI will be held in parallel, June 1989. CGM will be processing the results of the DAD ballots on CGM Add.1 and GKS Add.1 and the PDAD ballots on CGM Add.2 and GKS Add.2. CGI will be processing the 2nd DP ballot on CGI.

5.7 Liaison with SC18

On 2nd July there was a liaison meeting between SC24 and SC18, lasting from 9:00 to 3:30. Although this was SC24 liaison, not junst CGM or WG3 liaison, some of the results are important for the future work on metafiles within SC24/WG3. Other reports from Tucson will present the results of this meeting more comprehensively.

Both SCs presented their structure and program of work, highlighting significant work in progress and work planned. SC24 gave a presentation on SPDL, on the ISO 8613 (ODA/ODIF) Color Addendum, on ISO 9541 (Font Architecture). From the standpoint of CGM and extended CGM, the most important result was the beginning of a dialog on how to incorporate the Font and Color work of SC18 into SC24 standards. Conversely, it appears that some SC24 work (e.g., CGI datastreams, raster, and other areas) may be of value to SC18 projects in progress. Refer to other Tucson reports for details of the meeting

and resolutions.

6. About Addendum 3

The Metafile R.G. did not deal with "Addendum 3", CGM extensions for technical drawing, publishing, and graphic art quality pictures. At Blakeney the position was taken that the work is important and should be progressed, possibly by a group comprised of experts from metafiles, CGI, APIs, etc.

WG1 recommended establishment of a number of study groups, some on specific technical topics (product data geometry, text, etc) and some on particular new standards efforts — see the WG1 report for comprehensive discussion. One of the study groups is for "CGM extensions for enhanced static picture capture" — "addendum 3". Lofton Henderson is the rapporteur. There will be at least two meetings in the next year, scheduled in conjunction with the technology groups' meetings. The first is tentatively scheduled for Germany, December, meeting in conjuction with geometric and text groups.

The U.S. should prepare positions on both scope & goals and technical content for this initial meeting.

7. Status and Schedules

Both CGM Add.1 and GKSM (GKS Add.1) will be forwarded for DAD ballot. Both CGM Add.2 and GKSM-3D (GKS Add.2) will be forwarded for PDAD ballot. The tentative schedules are:

CGM Addendum 1 and GKS Addendum 1:

- Oct 88: DAD ballot commences.
- Apr 89: DAD ballot closes.
- Jun 89: Editing meeting.
- Aug 89: IS text forwarded to ISO Central Secretariat.

CGM Addendum 2 and GKS Addendum 2:

- Nov 88: PDAD ballot commences.
- Feb 89: DAD ballot closes.
- Jun 89: Editing meeting.
- Aug 89: DAD text produced.
- Oct 89: DAD ballot commences.
- Apr 90: DAD ballot closes.
- Jun 90: IS text forwarded to ISO Central Secretariat.

This page left intentionally blank.

APPENDIX 10

TUCSON RESOLUTIONS

This page left intentionally blank.

RESOLUTION 1—Appointment of SC24 Convenors

SC24 approves the appointment of these Working Group Convenors for a three-year term of office:

WG Number	Title	Convenor
WG1	Architecture	P. Bono
WG2	Application Programming Interface	J. Bettels
WG3	Metafile and Device Interface	D. Arnold
WG4	Language Bindings	B. Shepherd
WG5	Validation, Testing, and Registration	B. Kirsch

RESOLUTION 2—SC24 Structure and Terms of Reference

SC24 approves the working group structure and terms of reference in document SC24 N213.

RESOLUTION 3—SC24 Procedures for Preparing NWI Proposals

SC24 instructs its Secretariat to forward SC24 N171 for 3 month letter ballot. The comments will be resolved by the Requirements RG at the WG1 meeting in March 1989. WG1 will report to the SC24 Chairman whether the comments have been satisfactorily resolved and will recommend to the Chairman whether the document (as revised) should be accepted by the Chairman on behalf of SC24 or circulated for another ballot.

RESOLUTION 4—Implementation of Requirements Procedures

SC24 requests its National Bodies and experts to assist its Working Group 1 in the implementation and maintenance of the procedures as described in Document SC24 N171 as revised by the results of the balloting period described in Resolution 3. National Bodies and experts are encouraged to cooperate in this endeavour and in finding sponsors for computer-aided assistance of this process.

RESOLUTION 5—Establishment of SC24 Advisory Group (AG)

SC24 establishes an Advisory Group which consists of one delegate appointed by each P-member's National body, SC24 Working Group Convenors, SC24 Chairman and Secretariat. The SC24 Chairman presides as Chair of the Advisory Group.

RESOLUTION 6—SC24 Advisory Group - Area of Work

The SC24 Advisory Group provides advisory and management support for the SC24 Chairman and Secretariat. The AG is also concerned with coordination of liaisons with Organizations outside SC24 (e.g. SC2, SC18, SC21, SC22, TC184, TC10) on all topics of concern to SC24. The AG helps formulate the Strategic Plan for SC24 and gives procedural guidance to Working Groups.

RESOLUTION 7—SC24 Advisory Group—Terms of Reference

SC24 AG should operate consistently with SC24 procedures and resolutions and shall report in writing to SC24 Plenary Meetings.

RESOLUTION 8—SC24 Advisory Group—National Point of Contact

SC24 requests its National Bodies to nominate a national point of contact for the SC24 AG in writing to the SC24 Secretariat by 1988-10-01. This will facilitate direct contact with the AG participants and aid continuity in AG membership.

RESOLUTION 9—Approval of Study Periods

SC24 approves the commencement of study periods to provide the basis for developing a consistent set of NWIs for the following new areas of work:

New APIs (resulting from the technical work of the GKS Review RG) containing functionality required in the 1993 time frame aimed at producing at least a NWI for the GKS user community.

Windowing Environments

An API for Imaging

Extensions to the CGM Static Picture Capture Capabilities.

Extensions to PHIGS

Each study period should consider at least these items in conjunction with the indicated Group:

Requirements	[WG1]
Reference Model	[WG1]
Encodings	[WG3]
Language Bindings	[WG4]
Registration	[WG5]
Validation and Testing	[WG5]

RESOLUTION 10—Approval of Study Groups

SC24 appoints Special Rapporteurs to study the following topics according to the Terms of Reference in the indicated documents:

Topic	Source	Terms of Reference
Improved Graphical Text Model	WG1	SC24 N172
Impact of Windowing on Graphics Standards	WG1	SC24 N174
Improved Graphical Input Model	WG1	SC24 N173
Product Data Geometry	WG1	SC24 N175
Extensions to PHIGS	WG2	SC24 N211

RESOLUTION 11—SC24 Program of Work

SC24 approves the items of work, target dates, and document editors in SC24 N187.

RESOLUTION 12—Appointment of SC24 Rapporteurs

SC24 approves these rapporteurs:

Rapporteur	Topic
G. Grinstein	Requirements
J. Poller	Reference Model of Computer Graphics
J. McConnell	GKS-3D
J. Bettels	PHIGS
K. Vecchiet	CGI
E. Moeller*	CGM
J. Pink	Validation and Testing
M. Skall	Registration
K. Brodlie	GKS Maintenance
J. Rix	New APIs for Computer Graphics
P. ten Hagen	Windowing Environments
G.S. Carson	Imaging API
L. Henderson	Extensions to CGM for enhanced static picture capture
H. Stenzel	Improved Graphical Text Model
W. Hübner	Impact of Windowing on Graphics Standards
R. van Liere	Improved Graphical Input Model
E. Jungmann	Product Data Geometry
A. Francis	Metafile Maintenance Rapporteur
W. Clifford (ad hoc)	Extensions to PHIGS
E. Moeller*	CGM for GKS Workstation audit trail

* temporary

RESOLUTION 13—SC24 External Liaisons

SC24 approves the following external liaison representatives:

Name	Liaison to	Subject
G.S. Carson	SC21	Open Systems
R. Fairbairns	SC21[WG5]	FTAM document types for CGM
D. Arnold	SC21[WG5]	Terminal Management
L. Kotsch J. McConnell	SC18	Text and Office Systems including SPDL
J. Rix	TC184/SC4	Industrial Automation Systems/ Standard for the Exchange of Product Model Data
A. Ducrot	SC2	Character Sets and Coding
E. Jungmann	JTC1 SG-FS	Functional Standardization
G. Schaeffer	SC22	C Language
M. Sparks	SC22	Language Binding Techniques
G. Cuthbert	SC22	Ada Language
I. Grieger	SC22	FORTRAN language
D. Larson	SC22	Pascal Language

RESOLUTION 13A—Advisory Group Meetings

SC24 approves the following meeting for its AG:

Date	Place	Topic	Category	Purpose
1989-03	FRG (Schönhut)	[adjacent to WG1]	AG	

RESOLUTION 14—Working Group 1 Ad Hoc/Rapporteur/Editing Meetings

SC24 approves the following schedule of meetings for its Working Group 1 for the period up to the next SC24 plenary meeting:

Dates	Place	Topic	Category	Purpose
88-11	Boston, USA (Grinstein)	Imaging API Requirements	RG RG	Evaluate National Body Input - set drafting tasks
88-11	Copenhagen Denmark (Korn)	Windowing Environment Impact of Windowing SG	RG RG	Evaluate National Body Input - set drafting tasks
88-12	FRG (Jungmann)	CGM Extensions Improved Text SG Product Data Geometry SG	RG RG RG	Evaluate National Body Input - set drafting tasks
89-01	Paris, France (Ducrot)	Reference Model of CG	RG	Produce 2nd Working Draft
between 88-12 and 89-02	??, UK (Brodlie)	GKS Maintenance VT & R Liaison (with WG5)	RG	Produce NWI and base document
??89-02	Amsterdam, NL (ten Hagen)	New API Improved Input SG	RG RG	Evaluate National Body Input - set drafting tasks
??89-03	FRG (Encarnaçao)	Requirements ??	RG WG1	Consider National Body Comment on SC21 N171 and revise
??89-04	FRG (Krömker)	Imaging API	RG	Evaluate National Body Input - set drafting tasks

89-05	Copenhagen Denmark (Korn)	Requirements	RG	Planning for implementation of N171. Review forms. Begin processing requirements.
??89-04	USA Skall	Product Data Geometry SG [Joint meeting with CGM extensions]	RG	Agree output Documents
??89-06	Amsterdam, Netherlands (ten Hagen)	Windowing Environments Impact of Windowing SG	RG RG	Agree output Documents
??89-07	??, UK (Cartledge)	New API	RG	Examine new API in light of other study groups.
		Improved Input SG	RG	Agree output Documents
??89-08	FRG (Poller)	Reference Model	RG	Take 2nd WD comments.
		GKS Maintenance	RG	Revise base documents in light of National Body comments
89-07	Boston, USA (Grinstein)	Requirements	RG	Examine and reevaluate implementation of N171. Review and revise forms. Process requirements.
??89-09	??,Japan (Kawai)	Imaging API	RG	Agree Output Document

RESOLUTION 15—Working Group 2 Ad Hoc/Rapporteur/Editing Meetings

SC24 approves the following schedule of meetings for its Working Group 2 for the period up to the next SC24 plenary meeting:

Date	Place	Topic	Category	Purpose
between 88-10 and 88-11	FRG (Rix)	PHIGS Extn.	ad hoc RG	Prepare init. draft
between 89-01 and 89-02	Switzerland (Bettels)	PHIGS Extn.	ad hoc RG and WG	Study SC24 ballot results and revise NWI proposal for JTC1 letter ballot
between 89-06 and 89-07	UK (Stapleton)	PHIGS Extn.	RG	Study comments on Initial draft and prepare PDAD

RESOLUTION 16—Working Group 3 Ad Hoc/Rapporteur/Editing Meetings

SC24 approves the following schedule of meetings for its Working Group 3 for the period up to the next SC24 plenary meeting:

Dates	Place	Topic	Category	Purpose
19-23 Sept. 1988	Norfolk, UK (D. Arnold)	CGI	Drafting Meeting	To implement resolutions of issues from Valbonne, San Diego, and Tucson CGI RG meetings and preparation of 2nd DP text.
May 1989	Berlin/St. Augustin FRG (C. Egelhaff)	CGI [Liaison with VTR]	Pre-Meeting	Preliminary analysis of comments on the 2nd DP
June /July 1989	Hawaii USA (D. Larson)	CGM Add GKS Audit Trail	Editing Meeting	To prepare responses to CGM DAD1 ballot and to prepare final CGM AD1 text; to review comments on CGM PDAD2 ballot; to prepare responses to GKS DAD1 ballot and to prepare final GKS AD1 text.
		CGI	Editing Meeting	To prepare responses to 2nd DP ballot and to prepare DIS text.
		WG 3	Ad Hoc	To resolve joint problems with CGI/CGM addenda drafting.

RESOLUTION 17—Working Group 4 Ad Hoc/Rapporteur/Editing Meetings

SC24 approves the following schedule of meetings for its Working Group 4 for the period up to the next SC24 plenary meeting:

Dates	Place	Topic	Category	Purpose
Sept. 13-15 1988	Ft. Collins USA	Address comments on: DIS PHIGS/FORTRAN	Editing Meeting	Produce IS PHIGS/FORTRAN
Nov. 11-12 1988	Amsterdam Netherlands	Address comments on: DP PHIGS/C & Pascal WD GKS-3D/C & WD GKS/C	Pre-meeting	Prepare draft responses on PHIGS/C & Pascal GKS-3D/C & GKS/C
Nov. 14-15	Amsterdam Netherlands	Address comments on: DP PHIGS/C & Pascal	Editing Meeting	Recommend DIS PHIGS/C & Pascal
Nov. 16-21 1988	Amsterdam Netherlands	comments on: WD GKS-3D/C, Ada WD GKS/C	WG	Recommend dp GKS-3D/C & Ada GKS/C
March 1989	Sheffield UK	Address comments on: GKS-3D FORTRAN	Pre-meeting	Prepare draft responses on GKS-3D/FORTRAN
March 1989	Sheffield UK	Address comments on: GKS-3D FORTRAN	Editing Meeting	Produce IS GKS-3D FORTRAN
March 1989 [adjacent to WG1 and AG]	Sheffield UK	Address comments on: GKS-3D Pascal	WG	Recommend DP GKS-3D/Pascal
June 6-8 1989	Melbourne USA	Address comments on: PHIGS/Ada	Pre-meeting	Prepare draft responses on PHIGS/Ada
June 9 1989	Melbourne USA	Address comments on: PHIGS/Ada	Editing Meeting	Produce IS PHIGS/Ada

June 9-10 1989	Melbourne USA	Address comments on: GKS/C, GKS-3D/C & Ada	Editing Meeting	Recommend DIS GKS/C, GKS-3D/Ada&C
June 12-15 1989	Melbourne USA	Address comments on: CGI/C & FORTRAN	Ad Hoc WG	Register DP CGI/C&FORTRAN

RESOLUTION 18—SC24 Documents for Comment and 3 Month Letter Ballot

SC24 instructs its Secretariat to circulate the following documents for a 3 month letter ballot:

Document	Title	Comment Processing
SC24N171	Procedures for preparing NWI proposals	Comments resolved at Requirements meeting, FRG, March, 1988
SC24N211Rev SC24 N224	NWI proposal for Extensions to PHIGS	Comments for WG2 meeting Switzerland, Jan or Feb 89.
SC24N227	NWI proposal for GKS Maintenance	Comments for GKS Maintenance RG between 88-12 and 89-12
SC24 N176Rev	GKS Maintenance NWI Proposal	Comments for GKS Maintenance RG between 88-12 and 89-12

RESOLUTION 19—Working Group 5 Ad Hoc/Rapporteur/Editing Meetings
SC24 approves the following schedule of meetings for its Working Group 5 for the period up to the next SC24 plenary meeting:

Dates	Place	Topic	Category	Purpose
between Dec 88 and Feb 89	UK (Pink)	Registration Validation and Testing	WG	Review comments. Produce DP Text liaison GKS maint Study PICS applicabiiity Study profile applicability
[Along with GKS maintenance (WG1)]				
June - July 1989	??, FRG (Kirsch)	VTR	WG	Review comments. Review reg. proposals

The WG5 meeting between December 88 and February 89 will be held in parallel with the WG1 GKS maintenance rapporteur group meeting.

RESOLUTION 20—New Structure for the Next Generation of Graphics Standards

SC24 believes that the current way of producing standards will not lead to coordinated, integrated, timely standards with international scope and influence. SC24 instructs its Working Group 1 to investigate the feasibility of adopting a component/framework structure, as described in SC24 N139 for its next generation of graphics standards and notes that the adoption of this structure may require a new development process.

RESOLUTION 21—Deadline for Finishing Semantic Standards

SC24 resolves that the CGI project not be subject to restructuring under the new development process, unless CGI Parts 1-6 fail to progress as planned, with editing decisions for DIS texts taken before the 1989 SC24 plenary meeting.

RESOLUTION 22—GKS Maintenance

SC24 approves the following procedure and schedule for GKS Maintenance:

1) A draft NWI proposal (SC24 N227) will be circulated for National Body comment, with comments specifically being directed at the extent to which extensions to GKS functionality shall be included in the next revision of GKS. The guidelines contained in Document SC24 N176 should be observed when commenting on the draft NWI.

2) A meeting to resolve the comments and to prepare a final NWI and base document will be convened by Ken Brodlie and held in the UK between 1988-12-01 and 1989-02-28.

3) Subject to the approval of the SC24 Chair and the WG1 Convenor, the NWI will be submitted to JTC1 for ballotting and assignment of the project to SC24/WG2.

4) Simultaneously with step 3, the SC24 Secretariat will issue a conditional ballot to register the base document as a DP.

5) Subsequent to DP registration, the SC24 Secretariat will issue a DP ballot on the DP text.

6) Upon the close of the DP ballot, the SC24 Secretariat will schedule an Editing Meeting to discuss the comments received on the DP ballot and to prepare a disposition of comments and a new text.

RESOLUTION 23—SC24 Documents for Study and Comment

SC24 instructs its Secretariat to circulate the following documents for study and comment. Comments are to be submitted as indicated below:

Doc. No.	Title	Comments To Whom	Comments Due
SC24 N177	Working Draft Reference Model for Computer Graphics	SC24 Secr. SC24/WG1 Ref. Model Rapp.	1988-10-30
SC24 N178	The Use of Component/ Framework Description Techniques in the Specification of Computer Graphics Standards	SC24 Secr. WG1 Conv.	1988-10-30
SC24 N179	Sample Forms Used to Acquire and Record Requirements	SC24 Secr. SC24/WG1 Requirements Rapporteur	1988-10-30
SC24 N175	Terms of Reference for Product Data Geometry	SC24 Secr. Product Data Special Rapporteur	1989-01-15
SC24 N185	Working Draft Conformance Testing of Implementations of Graphics Standards	SC24 Secr. WG5 Conv. VT Rapp.	1988-11-30
SC24 N209	CGI character encoding ID	SC24 Secr. SC24/WG3	1989-03-01
SC24 N210	CGI binary encoding ID	SC24 Secr. SC24/WG3	1989-03-01
SC24 N 180	GKS/C WD with the changes agreed in Tucson.	SC24 Secr. SC24/WG4	1988-10-31 for Amsterdam
SC24 N 181	GKS-3D/C WD with the changes agreed in Tucson.	SC24 Secr. SC24/WG4	1988-10-31 for Amsterdam.
SC24 N 189	GKS-3D/Ada WD with the changes agreed in Tucson.	SC24 Secr. SC24/WG4	1988-10-31 for Amsterdam

SC24 N 190	GKS-3D/Pascal WD with the changes agreed in Tucson.	SC24 Secr. SC24/WG4	1988-12-31 for Sheffield
SC24 N 191	CGI/C WD with the changes agreed in Tucson.	SC24 Secr. SC24/WG4	1989-04-15 for Melbourne
SC24 N 192	CGI/FORTRAN WD with the changes agreed in Tucson.	SC24 Secr. SC24/WG4	1989-04-15 for Melbourne
SC24 N 193	List of Approved Abbreviations for Bindings	SC24 Secr. SC24/WG4	1989-04-01
SC24 N 194	CGI Language Bindings issues list	SC24 Secr. SC24/WG4	1989-04-01
SC24 N 223	Applicability of Static Picture Capture Files to GKS-3D and PHIGS	SC24 Secr. SC24/WG3	1989-04-01
SC24 N229	Reponsibilities of Liaison Officers	SC24 Secr.	1988-11-30
SC24 N228	Explanation of Test Requirements	SC24 Secr. SC24/WG5	1988-11-30

RESOLUTION 24—Registration Letter Ballot

SC24 instructs its Secretariat to circulate a 90 day letter ballot on document SC24 N226 regardless of the status of ISO/TR 9973 (SC24 N132) .

Doc. No.	Title	Comments To Whom	Comments Due
SC24 N184	Registration Proposals	SC24 Secr. WG5 Conv. Reg. Rapp.	1988-11-30

RESOLUTION 25—SC24 Meeting Plan for 1989 Plenary

SC24 approves the following meeting schedule for its WGs in conjunction with SC24 Plenary in Brazil in 1989:

X = plenary

		M	T	W	T	F	S	S	M	T	W	T	F	S	
SC24 Plenary	(2 days)												X	X	
Advisory Group	(1½ hours)	X	X		X			X		X		X			
WG1: Plenary	(4)		X	X	X	X									
WG2: Plenary	(1/2)											X			AM
GKS Maint.	(3)								X	X	X				
WG3: Plenary	(1½)			X*							X				
CGM Add (GKS ADD)	(5)			X	X			X	X	X					
CGI	(5)			X	X			X	X	X					
CGM Maintenance	(1)									X					
WG4: Plenary	(1/2)										X				PM
Working group	(3)			X	X			X	X	X		X			
WG5: Plenary	(1)										X				
VT	(3)			X	X			X							
R	(2)							X	X	X					

XX LaB Pre Meeting

X LaB Drafting

* Half day only

RESOLUTION 26—Liaison Statements

SC24 approves the following liaison statements and instructs its secretariat to forward them to the indicated liaison organizations:

Doc. No.	Title	To Whom
SC24 N208	Liaison Statement to SC18 Regarding SPDL	SC18
SC24 N198	Response to TC10 and UK Comments on Proposal for Registration of Graphical Items (ISO/IEC JTC1/SC24 N132)	TC10
SC24 N216	Response to SC18 N1383 (SC24 N118)	SC18
SC24 N 217	Comments on SC22 N466, Guidelines for Language Bindings.	SC22
SC24 N212	Liaison Statement to SC22	SC22

RESOLUTION 27—Documents for DP Ballot

In recognition of the substantial progress made with the CGI at the Tucson meeting SC24 instructs its Secretariat to circulate the text resulting from the September 1988 CGI Drafting meeting for 2nd DP ballot. The Document Editor is instructed to make the text resulting from the September 1988 Drafting meeting available to the SC24 Secretariat by 15 November 1988.

RESOLUTION 28—Structure of Work on 2D Metafiles

Whereas it was decided at the SC21/WG2 meeting at Egham in 1986 that a GKSM should be produced in a timely fashion by an addendum to CGM, and

whereas the Metafile Rapporteur Group of WG3 has substantially completed the technical work including the extensions for both a static picture capture metafile and a GKS metafile to replace annex E of GKS IS 7942, and

whereas the Metafile Rapporteur Group believes for the reasons detailed in SC24 N154, SC24 N156 and SC24/WG3 N32, that there are technical problems with continuing to progress the GKS Metafile portion of Addendum 1 as an addendum to CGM,

therefore SC24 directs WG3 to convert the GKS Metafile portion of Addendum 1 from an addendum to CGM to an addendum containing a non-normative annex to GKS IS 7942 and if possible progressed in the same timescale as CGM addendum 1. Further, this should become a normative annex to GKS during the maintenance process of

GKS.

RESOLUTION 29—Structure of Work on 3D metafiles

Whereas it was decided at the SC21/WG2 meeting at Egham in 1986 that a GKS-3D metafile should be produced in a timely fashion by an addendum to CGM, and

whereas the Metafile Rapporteur Group of WG3 has substantially completed the technical work including the extensions for both a 3D static picture capture metafile and a GKS-3D metafile to replace annex E of GKS-3D ISO/IEC DIS 8805, and

whereas the Metafile Rapporteur Group believes for the reasons detailed in SC24 N154, SC24 N156 and SC24/WG3 N32, that there are technical problems with continuing to progress the GKS 3-D Metafile portion of Addendum 2 as an addendum to CGM.

Therefore SC24 directs WG3 to convert the GKS-3D Metafile portion of Addendum 2 from an addendum to CGM to an addendum containing a non-normative annex to GKS-3D, ISO/IEC DIS 8805 and progressed in the same timescale as CGM addendum 2.

RESOLUTION 32—Withdrawal of DP Registration for GKS/C

SC24 withdraws the approval for DP registration for GKS/C (SC21 N1413) because the process of converging GKS/C with other GKS-3D and PHIGS bindings has produced an almost totally new document which is being recommended for Working Draft circulation.

RESOLUTION 34—SC24 Delegation of Authority

SC24 approves the delegation of authority to SC24/WG4 to forward documents for registration as draft proposals and instructs its Secretariat to register the following documents for DP letter ballot:

Document Title	Meeting to register
GKS-3D/Ada	November 1988 meeting in Amsterdam
GKS-3D/C	November 1988 meeting in Amsterdam
GKS/C	November 1988 meeting in Amsterdam
GKS-3D/Pascal	March 1989 meeting in Sheffield, UK

CGI/C	June 1989 meeting in Melbourne, USA
CGI/FORTRAN	June 1989 meeting in Melbourne, USA

RESOLUTION 35—Prompt Circulation of Comments to Document Editors

SC24 instructs its secretariat to distribute copies of National Body comments to the editor of the impacted document as the comments are received.

RESOLUTION 36—Language Bindings for Registered Items

SC24 recognizes the value of correct language bindings for registered items.

SC24 requests that National Bodies shall not forward incomplete proposals, including missing language bindings, to the Registration Authority because SC24/WG4 is not required to originate language bindings for proposals when these bindings are missing.

RESOLUTION 37—SC24 Documents for DIS Registration

SC24 instructs its Secretariat to forward the following document to ISO Central Secretariat for registration as a Draft International Standard:

> The PHIGS binding to Ada (ISO 9593-3), with the changes agreed in Tokyo and discussion and resolution, in Tucson, of the issue contained in SC24/WG4/N011. DIS text will be provided to SC24 Secretariat in September 1988, following validation of the resolution of the last issue.

RESOLUTION 38—Simultaneous Development of Semantic Standards and Language Bindings

SC24 adopts the following policy:

> That all functional specifications which reach the stage of DIS or IS should be accompanied by at least one language binding or encoding for that functional specification. Specifically, if the functional specification is at the stage of DIS, at least one language binding or encoding should be at the stage of DP, and if the functional specification is to become an IS, at least one language binding or encoding should be at the stage of DIS.

RESOLUTION 39—Progression of PHIGS C and PHIGS Extended Pascal

SC24 instructs its Secretariat to check that the C language standard is at DIS stage before registering the PHIGS C Language Binding as a DIS, and that the Extended Pascal language standard is at DIS stage before registering the PHIGS Extended Pascal Language Binding as a DIS.

RESOLUTION 40—Conformance Testing Standard Progression

SC24 approves the progression of project JTC1.24.7 as a single part standard. The standard will contain general concepts and guidelines for conformance testing of the complete range of graphics standards. Specific details for each standard will be produced in a Test Requirements document for each standard.

RESOLUTION 42—Conformance Clauses

SC24 approves the following procedures for developing and reviewing the conformance sections of Semantic Standards developed within SC24:

1. The current draft of the Conformance Testing Standard (project JTC1.24.7) should be consulted.

2. The SC24/WG5 Convenor and the Validation and Testing Rapporteur should be notified of the place and time of discussions relating to the Standard conformance sections and WG5 experts should be invited to participate in these discussions.

3. The text for the conformance sections shall be developed jointly by the WG5 Validation and Testing Rapporteur Group and the Standard Group.

RESOLUTION 43—PICS Proforma

SC24 instructs its Working Group 5 to investigate the use of PICS (Protocol Implementation Conformance Statement) proformas to assist in the development of conformance requirements and the application of conformance test suites for graphics standards. WG5 is instructed to produce a report by 15 March 1989 for circulation to SC24 National Bodies for comment by 15 July 1989. The SC24 Chair will write to the SC21 Chair to request SC21 participation in this work.

RESOLUTION 44—Application Profiles

SC24 instructs its Working Group 5 to investigate the applicability of Application and Constituency Profiles and their relationship to graphics standards and registration. In particular, WG5 should examine whether the profiles can assist in the definition of conformance requirements and consider if SC24 should be involved in the

standardisation or registration of profiles. The SC24 Chair will write to the SC21 Chair to request SC21 participation in this work.

RESOLUTION 46—Appreciation to GKS Binding editors

SC24 unanimously expresses its appreciation to D. Larson, editor of GKS Pascal (ISO/IEC 8651-2) and J. McConnell, editor GKS FORTRAN (ISO/IEC 8651-1) on publication of the standard and completion of their work.

RESOLUTION 47—SC24 Five Year Meeting Plan

SC24 approves the following five year plan for its plenary meetings and notes that SC24 meetings are held at least each 18 months.

Dates	Place
15 Oct. - 15 Nov 1989	Iguassu Falls, Brazil
April 1991	U.K.
October 1992	Germany, F.R.
April 1994	France

RESOLUTION 48—EEC Standardisation Activities Within JTC1/SC24 Scope of Work [Denmark, FRG]

Whereas

1) SC24 is aware that the EEC Commission is developing standard activities which are thoroughly related to the scope of work of SC24 such as:
   - GPOS (PPSC-IT N252), (Generalized Portable Operating System);
2) the European Standards (EN) have a higher priority for EEC and EFTA countries,

SC24 offers to cooperate and assist in developing the on-going EEC standard activites, with regard to the SC24 area of work (Computer Graphics).

RESOLUTION 50—SC24 Appreciation for Meeting

SC24 unanimously expresses its appreciation to the meeting organizers and contributors:

Organization	Peter R. Bono Associates, Inc.
Secretarial Support	Susan Bonde, Diane Bono, Elaine Bono, Brenda Carson, Gillian Hall
Financial Support	National Computer Graphics Association Advanced Technology Center Apollo Computer Digital Equipment Corporation Hewlett-Packard Company International Business Machines Lockheed/CalComp Corporation National Semiconductor Corporation Pansophic Systems, Inc. Precision Visuals, Inc. Tektronix, Inc. Unisys Corporation
Computer Support	Chin Associates Geri Cuthbert GSC Associates Inc. Sun Microsystems, Inc.

SC24 unanimously expresses appreciation to its Chair, Dr. Jürgen Schönhut, the Secretariat, and the Drafting Committee (G.S. Carson, C. Cartledge, I. Korn, and J. Rix) for their work in support of the meeting.

RESOLUTION 51—Appreciation to Janet Chin

SC24 unanimously expresses their appreciation to Janet Chin for years of unstinting effort as leader of the US delegation.

RESOLUTION 52—Language Binding Issues Librarians

SC24 notes that WG4 has established the position of issues librarian for each language binding. The duties of issues librarians are to record issues, including their arguments and resolutions, and maintain and circulate the library. SC24 directs its Working Groups to identify to the WG4 convenor those individuals from their membership who can participate in the activities of WG4 in this role, on a part time basis.

RESOLUTION 53—SC24 Documents for DP/PDAD registration

SC24 instructs its Secretariat to circulate the following documents for parallel PDAD registration ballot and conditional PDAD ballot:

Document no.	Title	Condition
SC24 N219	CGM Addendum 2, part 1	As amended by decisions taken at Tucson
SC24 N220	CGM Addendum 2, part 2	As defined by decisions taken at Tucson
SC24 N221	CGM Addendum 2, part 3	As defined by decisions taken at Tucson
SC24 N222	CGM Addendum 2, part 4	As defined by decisions taken at Tucson

RESOLUTION 54—SC24 Documents for DIS/DAD registration

SC24 instructs its Secretariat to forward the following documents to the JTC1 Secretariat for DAD ballot:

Document no.	Title	Condition
SC24 N230	CGM Addendum 1, part 1	As amended by decisions taken at Tucson
SC24 N231	CGM Addendum 1, part 2	As amended by decisions taken at Tucson
SC24 N232	CGM Addendum 1, part 3	As amended by decisions taken at Tucson
SC24 N233	CGM Addendum 1, part 4	As amended by decisions taken at Tucson
SC24 N234	GKS Addendum 1	To be derived from all parts of CGM addendum 1, following the decisions taken at Tucson

RESOLUTION 55—SC24 Document for Registration Sponsorship

SC24 directs its Secretariat to circulate the registration proposals in Document SC24 N225 for a 3 month letter ballot to approve SC24 sponsorship of them for registration as graphical items.

RESOLUTION 56—Extended Scope of the Register of Graphical Items

SC24 directs the WG5 Registration Rapporteur Group to consider ways in which the registration procedures can be extended to cover the additional graphical items required for registration by SC24 standards.

NIST-114A
(REV. 3-89)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER
NISTIR 4316

2. PERFORMING ORGANIZATION REPORT NUMBER

3. PUBLICATION DATE
MARCH 1991

4. TITLE AND SUBTITLE
A Collection of Technical Studies Completed for the Computer-Aided Acquisition and Logistic Support (CALS) Program Fiscal Year 1988 Volume 2 of 3, Graphics, CGM MIL SPEC

5. AUTHOR(S)
Roy S. Morgan, Editor

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)
U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
GAITHERSBURG, MD 20899

7. CONTRACT/GRANT NUMBER

8. TYPE OF REPORT AND PERIOD COVERED
NISTIR 10/87 - 9/88

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)
Office of the Secretary of Defense
Production and Logistics/Systems/CALS
Room 3B322, Pentagon
Washington, D.C. 20301-8000

10. SUPPLEMENTARY NOTES

[ ] DOCUMENT DESCRIBES A COMPUTER PROGRAM; SF-185, FIPS SOFTWARE SUMMARY, IS ATTACHED.

11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

Computer-aided Acquisition and Logistic Support (CALS) Program is a DoD Industry strategy to transition from paper-intensive acquistion and logistic processes to a highly automated and integrated mode of operation for the weapon systems of the 1990s. These volumes document the accomplishments of the National Institute of Standards and Technology to advance the development of technology and standards in support of CALS. These reports are divided into three volumes: 1, Text, Security, and Data Management; 2, Graphics, CGM MIL-SPEC; and 3, Graphics, CGM Registration.

Volume 2. Graphics: Progress in the Computer Graphics Metafile standard is described, including work in the graphics standards committees and the expansion and updating of the CALS CGM application profile. A draft Military Specifications for CGM is included. A plan for Extended CGM is presented, including documentation of relevant standards committee work.

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)
CGM; Extended CGM; graphics; graphics metafile; metafile

13. AVAILABILITY

[X] UNLIMITED

[ ] FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).

[ ] ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.

[X] ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.

14. NUMBER OF PRINTED PAGES
482

15. PRICE
A21

ELECTRONIC FORM